File created: 14-Sep-2023 23:40:42 **{WMEDLEY}<sources>INSPECT.;28**

edit by: rmk

changes to: (FNS INSPECTABLEFIELDNAMES)

previous date: 15-Jun-2023 16:03:17 {WMEDLEY}<sources>INSPECT.;27

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
;;
;; Copyright (c) 1982-1987, 1990-1991, 1993, 1995, 1999, 2018, 2021 by Venue & Xerox Corporation.

(RPAQQ INSPECTCOMS
       [(COMS ;; functions to implement an item window.  An ITEM window is a window that contains SELECTABLEITEMS.  An item from the
              ;; window is selected using the left button.  The middle button will then bring up a menu of commands that can be applyed to the
              ;; selected item.  An INSPECTW is a special type of ITEMW that maintains properties and their values for a datum.  It is used by the
              ;; inspector.
              (FNS INSPECTW.CREATE INSPECTW.REPAINTFN INSPECTW.REDISPLAY \INSPECTW.VALUE.MARGIN INSPECTW.REPLACE
                   INSPECTW.SELECTITEM \INSPECTW.REDISPLAYPROP INSPECTW.FETCH INSPECTW.PROPERTIES
                   DECODE.WINDOW.ARG DEFAULT.INSPECTW.PROPCOMMANDFN DEFAULT.INSPECTW.VALUECOMMANDFN
                   DEFAULT.INSPECTW.TITLECOMMANDFN \SELITEM.FROM.PROPERTY \INSPECT.COMPUTE.TITLE LEVELEDFORM
                   MAKEWITHINREGION)
              (FNS ITEMW.REPAINTFN \ITEM.WINDOW.BUTTON.HANDLER \ITEM.WINDOW.SELECTION.HANDLER
                   \INSPECTW.COMMAND.HANDLER ITEM.WINDOW.SET.STACK.ARG REPLACESTKARG IN/ITEM?
                   \ITEMW.DESELECTITEM \ITEMW.SELECTITEM \ITEMW.CLEARSELECTION \ITEMW.FLIPITEM PRINTANDBOX
                   PRINTATBOX ITEMOFPROPERTYVALUE)
              (FNS \ITEM.WINDOW.COPY.HANDLER \ITEMW.FLIPCOPY BKSYSBUF.GENERAL)
              (RECORDS SELECTABLEITEM)
              (VARS (MAXINSPECTARRAYLEVEL 300)
                    (MAXINSPECTCDRLEVEL 50)
                    MinSpaceBetweenProperyAndValue MaxInspectorPropertyValueWidth MaxValueLeftMargin
                    PropertyLeftMargin))
        (COMS                                                   ; functions for the inspector
              (FNS INSPECT \APPLYINSPECTMACRO INSPECT/BITMAP INSPECT/DATATYPE INSPECTABLEFIELDNAMES REMOVEDUPS
                   INSPECT/ARRAY INSPECT/TOP/LEVEL/LIST INSPECT/PROPLIST NONSYSPROPNAMES INSPECT/LISTP ALISTP
                   PROPLISTP INSPECT/ALIST ASSOCGET /ASSOCPUT INSPECT/PLIST INSPECT/TYPERECORD INSPECT/AS/RECORD
                   SELECT.LIST.INSPECTOR STANDARDEDITE NTHTOPLEVELELT SETNTHTOPLEVELELT DEDITE FINDRECDECL
                   FINDSYSRECDECL MAKE-INSPECTOR-PROFILE CONFIRM-SET)
              (GLOBALVARS INSPECTMACROS INSPECTALLFIELDSFLG INSPECTDONTSORTFIELDS SetPropertyMenu SetStackMenu
                     InspectMenu PropertyLeftMargin MaxValueLeftMargin INSPECTPRINTLEVEL InspectBitmapMenu
                     ItemWCommandMenu InspectPropsMenu MAXINSPECTARRAYLEVEL MAXINSPECTCDRLEVEL
                     MaxInspectorWindowWidth MaxInspectorWindowHeight INSPECT.HUNK.COMMANDS USERRECLST SYSPROPS
                     IT MinSpaceBetweenProperyAndValue MaxInspectorPropertyValueWidth)
              (INITVARS (INSPECTDONTSORTFIELDS T)
                     (INSPECTALLFIELDSFLG T)
                     (MaxInspectorWindowWidth 330)
                     (MaxInspectorWindowHeight 606))
              (VARS INSPECTPRINTLEVEL)
              ;; To deal with profiles in spawned processes
              (MACROS EVAL.AS.PROCESS.WITH.PROFILE WITH-INSPECTOR-ENV))
        (COMS                                                   ; Atom inspector
              (FNS INSPECT/ATOM SELECT.ATOM.ASPECT INSPECT/AS/FUNCTION SELECT.FNS.EDITOR))
        (COMS                                                   ; Compiled code inspector
              (FNS INSPECTCODE \TEDIT.INSPECTCODE \INSPECT/CODE/RESHAPEFN \INSPECT/CODE/REPAINTFN))
        (COMS                                                   ; Hash table inspector
              (FNS INSPECT/HARRAYP HARRAYKEYS INSPECTW.GETHASH INSPECTW.PUTHASH))
        [COMS                                                   ; Readtable, termtable inspectors
              (FNS RDTBL\NONOTHERCODES GETSYNTAXPROP SETSYNTAXPROP GETTTBLPROP SETTTBLPROP)
              (ADDVARS (INSPECTMACROS (READTABLEP RDTBL\NONOTHERCODES GETSYNTAXPROP SETSYNTAXPROP)
                              (TERMTABLEP (CHARDELETE WORDDELETE LINEDELETE RETYPE CTRLV EOL RAISE ECHOMODE
                                                      LINEDELETESTR 1STCHDEL NTHCHDEL POSTCHDEL EMPTYCHDEL ECHODELS?
                                                      CONTROL 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
                                                      22 23 24 25 26 27 28 29 30 31)
                                          GETTTBLPROP SETTTBLPROP]
        (COMS                                                   ; Hunk inspector
              (FNS INSPECT/AS/BLOCKRECORD INSPECT/TYPELESS LIST-ALL-BLOCKRECORDS INSPECT/HUNK
                   \INSPECT.DATATYPE.RAW.FETCH \INSPECT.FETCH.8 \INSPECT.FETCH.32 \INSPECT.FETCH.CHAR
                   \INSPECT.FETCH.FATCHAR \INSPECT.FETCH.PTR \INSPECT.STORE.8 \INSPECT.STORE.16
                   \INSPECT.STORE.32 \INSPECT.STORE.CHAR \INSPECT.STORE.FATCHAR \INSPECT.STORE.PTR
                   INSPECT/MAKE/CCODEP)
              (INITVARS (INSPECT.HUNK.COMMANDS '(("As 8-bit array" '(8 \GETBASEBYTE \INSPECT.STORE.8))
                                                 ("As 16-bit array" '(16 \GETBASE \INSPECT.STORE.16))
                                                 ("As 32-bit array" '(32 \INSPECT.FETCH.32 \INSPECT.STORE.32))
                                                 ("As Character array" '(8 \INSPECT.FETCH.CHAR
                                                                           \INSPECT.STORE.CHAR))
                                                 ("As Fat Character array" '(16 \INSPECT.FETCH.FATCHAR
                                                                                \INSPECT.STORE.FATCHAR])
```

;; functions to implement an item window. An ITEM window is a window that contains SELECTABLEITEMS. An item from the window is selected using

;; the left button. The middle button will then bring up a menu of commands that can be applyed to the selected item. An INSPECTW is a special type of ;; ITEMW that maintains properties and their values for a datum. It is used by the inspector.

```
(DEFINEQ
```

(INSPECTW.CREATE

```
  [LAMBDA (DATUM PROPERTIES FETCHFN STOREFN PROPCOMMANDFN VALUECOMMANDFN TITLECOMMANDFN TITLE SELECTIONFN WHERE
                PROPPRINTFN TAG)                                    ; Edited 12-Sep-2022 21:07 by rmk
                                                                    ; Edited  3-Sep-2022 23:48 by rmk
                                                                    ; Edited  5-Aug-87 09:52 by jop
    ;; Creates a window with an item list made up of properties and values
    (LET ((PROFILE (MAKE-INSPECTOR-PROFILE)))
         (WITH-INSPECTOR-ENV PROFILE (PROG [WINDOW VALUE PROPMENU VALUEMENU VALUEMARGIN SELITEMS MAXVALUEWIDTH
                                           (IWFONT (DEFAULTFONT 'DISPLAY))
                                           (PROPERTIESLST (COND
                                                              ((OR (NULL PROPERTIES)
                                                                   (LISTP PROPERTIES))
                                                               PROPERTIES)
                                                              (T
                                                               ; allow PROPERTIES to be a function
                                                                  (APPLY* PROPERTIES DATUM]
                                       (SETQ VALUEMARGIN (\INSPECTW.VALUE.MARGIN
                                                          (COND
                                                             (PROPPRINTFN (for PROP in PROPERTIESLST
                                                                             collect (APPLY* PROPPRINTFN PROP
                                                                                             DATUM)))
                                                             (T PROPERTIESLST))
                                                          IWFONT))
                                       (SETQ MAXVALUEWIDTH
                                        (COND
                                            (PROPERTIESLST (IMIN (IMAX (bind X for PROP in PROPERTIESLST
                                                                          largest (STRINGWIDTH (APPLY* FETCHFN
                                                                                                       DATUM
                                                                                                       PROP)
                                                                                               IWFONT T)
                                                                          finally (RETURN $$EXTREME))
                                                                       16)
                                                                  MaxInspectorPropertyValueWidth))
                                            (T                  ; no fields to inspect
                                               30)))
                                       (SETQ WINDOW (DECODE.WINDOW.ARG WHERE (IPLUS VALUEMARGIN
                                                                                    MAXVALUEWIDTH)
                                                                       (IMIN MaxInspectorWindowHeight
                                                                             (ITIMES (COND
                                                                                        (PROPERTIESLST (LENGTH
                                                                                                         PROPERTIESLST
                                                                                                         ))
                                                                                        (T 1))
                                                                                     (FONTHEIGHT IWFONT)))
                                                                       (\INSPECT.COMPUTE.TITLE TITLE DATUM)))
                                       (DSPFONT IWFONT WINDOW)
                                       (DSPRIGHTMARGIN 50000 WINDOW)
                                                          ; for now, can't handle multiple PROPCOMMANDFN output.  Put
                                                          ; right margin way out.
                                       (WINDOWPROP WINDOW 'DATUM DATUM)
                                                          ; initialize the properties of the window.
                                       (WINDOWPROP WINDOW 'STOREFN STOREFN)
                                       (WINDOWPROP WINDOW 'FETCHFN FETCHFN)
                                       (WINDOWPROP WINDOW 'PROPCOMMANDFN PROPCOMMANDFN)
                                       (WINDOWPROP WINDOW 'VALUECOMMANDFN VALUECOMMANDFN)
                                       (CL:WHEN TAG
                                           (SETQ TAG (CONCAT TAG ": "))
                                           (WINDOWPROP WINDOW 'INSPECTTAG TAG)
                                           [SETQ TITLE (CONCAT TAG (WINDOWPROP WINDOW 'TITLE]
                                           (WINDOWPROP WINDOW 'TITLE TITLE))
                                       (WINDOWPROP WINDOW 'INSPECTWTITLE TITLE)
                                       (WINDOWPROP WINDOW 'TITLECOMMANDFN TITLECOMMANDFN)
                                       (WINDOWPROP WINDOW 'SELECTIONFN SELECTIONFN)
                                       (WINDOWPROP WINDOW 'PROPERTIES PROPERTIES)
                                       (WINDOWPROP WINDOW 'PROPPRINTFN PROPPRINTFN)
                                       (WINDOWPROP WINDOW 'BUTTONEVENTFN (FUNCTION
                                                                             \ITEM.WINDOW.BUTTON.HANDLER))
                                       (WINDOWPROP WINDOW 'COPYBUTTONEVENTFN (FUNCTION
                                                                                 \ITEM.WINDOW.COPY.HANDLER))
                                       (WINDOWPROP WINDOW 'REPAINTFN (FUNCTION INSPECTW.REPAINTFN))
                                       (WINDOWPROP WINDOW 'SCROLLFN (FUNCTION SCROLLBYREPAINTFN))
                                   ;; when we create the window, record the read print environment so that the window methods  can
                                   ;; use the same one, rather than inheriting form the mouse process.
                                       (WINDOWPROP WINDOW 'PROFILE PROFILE)
                                       (RETURN (INSPECTW.REDISPLAY WINDOW NIL VALUEMARGIN])
```

(INSPECTW.REPAINTFN

```
  [LAMBDA (WINDOW REGION)                                           ; Edited  8-Apr-87 16:36 by jop
    ;; repaints the selectable items in (an inspect window.  This knows that the items are stored in increasing order.)
```

```
                                                        ;  restore the profile that was used when the inspector was
                                                        ; instantiated, so that packages, escapes etc. are the same.
    [WITH-INSPECTOR-ENV (WINDOWPROP WINDOW 'PROFILE)
           (COND
              [REGION                                   ; only clip to region if a region is given.
                     (PROG ((SELITEMS (WINDOWPROP WINDOW 'SELECTABLEITEMS))
                            (WREG (DSPCLIPPINGREGION NIL WINDOW))
                            LINEBASE SELECTABLEITEMREGION PROPPRINTFN)
                           (SETQ LINEBASE (fetch (REGION TOP) of WREG))
                      ABOVELP
                                                        ; skip those above the window.
                           (COND
                              ((NULL SELITEMS)
                               (RETURN))
                              ((IGREATERP (fetch (REGION BOTTOM) of (fetch (SELECTABLEITEM SELECTABLEITEMREGION)
                                                                           of (CAR SELITEMS)))
                                          LINEBASE)
                               (SETQ SELITEMS (CDR SELITEMS))
                               (GO ABOVELP)))           ; determine the bottom line base
                           (SETQ LINEBASE (fetch (REGION BOTTOM) of WREG))
                      PRINTLP
                                                        ; print them as long as they are visible.
                           (COND
                              ((NULL SELITEMS)
                               (RETURN))
                              ((IGREATERP [fetch (REGION PTOP) of (SETQ SELECTABLEITEMREGION (fetch (
                                                                                    SELECTABLEITEM

                                                                                 SELECTABLEITEMREGION
                                                                                       )
                                                                                 of (CAR SELITEMS]
                                          LINEBASE)     ; still possibly visible, check for horizontal fit before printing.
                               (COND
                                  ((REGIONSINTERSECTP REGION SELECTABLEITEMREGION)
                                   (PRINTATBOX [COND
                                                  [[AND (EQ (fetch (SELECTABLEITEM ITEMINFOTYPE)
                                                               of (CAR SELITEMS))
                                                            'PROPERTY)
                                                        (WINDOWPROP WINDOW 'DATUM)
                                                        (SETQ PROPPRINTFN (WINDOWPROP WINDOW 'PROPPRINTFN]
                                                        ; hook for property print functions Should be cleaned up.
                                                   (APPLY* PROPPRINTFN (fetch (SELECTABLEITEM ITEMINFO)
                                                                              of (CAR SELITEMS))
                                                           (WINDOWPROP WINDOW 'DATUM]
                                                  (T (fetch (SELECTABLEITEM ITEMINFO) of (CAR SELITEMS]
                                               WINDOW SELECTABLEITEMREGION)))
                               (SETQ SELITEMS (CDR SELITEMS))
                               (GO PRINTLP]
              (T                                        ; if no region, use other repaintfn to repaint them all.
                 (\ITEMW.REPAINTFN WINDOW]              ; if there is a selected item, flip it too in case some of it was in the
                                                        ; newly exposed area.
    (AND (WINDOWPROP WINDOW 'CURRENTITEM)
         (\ITEMW.FLIPITEM (WINDOWPROP WINDOW 'CURRENTITEM)
                WINDOW])
```

(INSPECTW.REDISPLAY

```
  [LAMBDA (WINDOW PROPS VALUEMARGIN)                    ; Edited  8-Apr-87 16:39 by jop
                                                        ; redisplays an itemw to get the newly updated fields.
    (COND
       [PROPS (COND
                 ((NLISTP PROPS)
                  (\INSPECTW.REDISPLAYPROP WINDOW PROPS))
                 (T (for PROP in PROPS do (\INSPECTW.REDISPLAYPROP WINDOW PROP]
       (T (WITH-INSPECTOR-ENV
              (WINDOWPROP WINDOW 'PROFILE)
              (PROG ((DATUM (WINDOWPROP WINDOW 'DATUM))
                     (PROPERTIES (INSPECTW.PROPERTIES WINDOW))
                     (FETCHFN (WINDOWPROP WINDOW 'FETCHFN))
                     (PROPCOMMANDFN (WINDOWPROP WINDOW 'PROPCOMMANDFN))
                     (VALUECOMMANDFN (WINDOWPROP WINDOW 'VALUECOMMANDFN))
                     (PROPPRINTFN (WINDOWPROP WINDOW 'PROPPRINTFN))
                     PROPERTY-PNAMES VALUE PROPMENU VALUEMENU SELITEMS)
                    (SETQ PROPERTY-PNAMES (COND
                                             (PROPPRINTFN (for PROP in PROPERTIES collect (APPLY* PROPPRINTFN PROP
                                                                                                  DATUM)))
                                             (T PROPERTIES)))
                    [SETQ VALUEMARGIN (OR VALUEMARGIN (\INSPECTW.VALUE.MARGIN PROPERTY-PNAMES (DSPFONT NIL WINDOW]
                                                        ; remove old selected item if any
                    (\ITEMW.DESELECTITEM NIL WINDOW)
                    (CLEARW WINDOW)
                    (WINDOWPROP WINDOW 'TITLE (\INSPECT.COMPUTE.TITLE (WINDOWPROP WINDOW 'INSPECTWTITLE)
                                                  DATUM WINDOW))
              ;; might be faster to only print and determine positions for the ones that are visible and keep track of which haven't been seen yet but
              ;; this is easier for now.
                    (MOVETOUPPERLEFT WINDOW (DSPCLIPPINGREGION NIL WINDOW))
```

```
                [WINDOWPROP WINDOW 'SELECTABLEITEMS
                       (SETQ SELITEMS
                        (for PROP in PROPERTIES as PROPNAME in PROPERTY-PNAMES
                           join (COND
                                   [PROPNAME (LIST (create SELECTABLEITEM
                                                          SELECTABLEITEMREGION _ (PRINTANDBOX PROPNAME WINDOW
                                                                                        PropertyLeftMargin)
                                                          COMMANDFN _ (OR PROPCOMMANDFN (FUNCTION
                                                                                           DEFAULT.INSPECTW.PROPCOMMANDFN
                                                                                           ))
                                                          ITEMINFO _ PROP
                                                          ITEMINFOTYPE _ 'PROPERTY)
                                                   (create SELECTABLEITEM
                                                          SELECTABLEITEMREGION _
                                                          (PRINTANDBOX (COND
                                                                          ((NLSETQ (SETQ VALUE
                                                                                         (APPLY* FETCHFN DATUM PROP)
                                                                                         ))
                                                                           VALUE)
                                                                          (T
                                                                 ; error during access.
                                                                             (SETQ VALUE "** error during
                                                                              access **")))
                                                          WINDOW VALUEMARGIN
                                                          MinSpaceBetweenProperyAndValue)
                                                          COMMANDFN _ (OR VALUECOMMANDFN (FUNCTION
                                                                                           DEFAULT.INSPECTW.VALUECOMMANDFN
                                                                                           ))
                                                          ITEMINFO _ VALUE
                                                          ITEMINFOTYPE _ (CONS PROP]
                                           (T                    ; if property name returns NIL, print value in middle
                                              (CONS (create SELECTABLEITEM
                                                            SELECTABLEITEMREGION _
                                                            (PRINTANDBOX (COND
                                                                            ((NLSETQ (SETQ VALUE (APPLY* FETCHFN DATUM
                                                                                                         PROP)))
                                                                             VALUE)
                                                                            (T
                                                                 ; error during access.
                                                                               (SETQ VALUE "** error during access **"))
                                                                            )
                                                            WINDOW
                                                            (LRSH VALUEMARGIN 1))
                                                    COMMANDFN _ (OR VALUECOMMANDFN (FUNCTION
                                                                                     DEFAULT.INSPECTW.VALUECOMMANDFN
                                                                                      ))
                                                    ITEMINFO _ VALUE
                                                    ITEMINFOTYPE _ (CONS PROP]
                [WINDOWPROP WINDOW 'EXTENT
                       (PROG [(NOWEXTENT (COND
                                           [SELITEMS (create REGION using (fetch (SELECTABLEITEM
                                                                                    SELECTABLEITEMREGION)
                                                                                  of (CAR SELITEMS]
                                           (T                    ; don't have any items;  make extent empty.
                                              (create REGION
                                                      LEFT _ 0
                                                      BOTTOM _ 0
                                                      WIDTH _ 0
                                                      HEIGHT _ 0]
                             (for SELITEM in (CDR SELITEMS) do (EXTENDREGION NOWEXTENT (fetch (SELECTABLEITEM

                                                                                              SELECTABLEITEMREGION
                                                                                               )
                                                                                             of SELITEM)))
                             (RETURN NOWEXTENT)))                ; limit scrolling so that it won't go off the top.
                [WINDOWPROP WINDOW 'SCROLLEXTENTUSE 'LIMIT)
                (RETURN WINDOW])
```

(\INSPECTW.VALUE.MARGIN

```
  [LAMBDA (PROPS FONT)                                           ; Edited  2-Feb-87 17:15 by jop
    ;; returns the x position in which the values of the properties should print.
    (IMIN (IPLUS (IMAX (MAXSTRINGWIDTH PROPS FONT T)
                       16)
                 MinSpaceBetweenProperyAndValue PropertyLeftMargin)
          MaxValueLeftMargin])
```

(INSPECTW.REPLACE

```
  [LAMBDA (INSPECTW PROPERTY NEWVALUE)                           ; Edited 22-Jun-87 17:43 by jop
    (PROG [(DATUM (WINDOWPROP INSPECTW 'DATUM))
           (STOREFN (WINDOWPROP INSPECTW 'STOREFN]
          (OR STOREFN (ERROR INSPECTW " does not have a STOREFN."))
          (OR DATUM (ERROR INSPECTW " doesn't have a DATUM"))
          [LET ((XCL:*EVAL-FUNCTION* 'CL:EVAL))
               ;; Use cl:eval, since it wouldn't choke on compiled closures
```

```
            (EXEC-EVAL (LIST STOREFN (KWOTE DATUM)
                             (KWOTE PROPERTY)
                             (KWOTE NEWVALUE]
        (RETURN (\INSPECTW.REDISPLAYPROP INSPECTW PROPERTY])
```

(INSPECTW.SELECTITEM

```
[LAMBDA (INSPECTW PROPERTY VALUEFLG)                    ; Edited 3-Feb-87 16:41 by jop
```

;; makes a selection in an inspect window. If another item is selected, it is deselected. If VALUEFLG is non-NIL, the value of the property is ;; selected, otherwise the property name is selected. If PROPERTY is NIL, any selected item is deselected and no item is selected. Returns the ;; previously selected item structure.

```
(PROG [(PREVIOUS (WINDOWPROP INSPECTW 'CURRENTITEM]
      (AND PREVIOUS (\ITEMW.DESELECTITEM PREVIOUS INSPECTW))
      (AND PROPERTY (\ITEMW.SELECTITEM (COND
                                          (VALUEFLG (ITEMOFPROPERTYVALUE PROPERTY INSPECTW))
                                          (T (\SELITEM.FROM.PROPERTY INSPECTW PROPERTY)))
                           INSPECTW))
      (RETURN PREVIOUS])
```

(\INSPECTW.REDISPLAYPROP

```
[LAMBDA (WINDOW PROPERTY)                               ; Edited 10-Apr-87 16:31 by jop
```

;; refetches and displays a property of an inspect window. This is called when a property has changed, to update the display.

```
(WITH-INSPECTOR-ENV (WINDOWPROP WINDOW 'PROFILE)
       (LET ((DATUM (WINDOWPROP WINDOW 'DATUM))
             (OLDVALUEITEM (ITEMOFPROPERTYVALUE PROPERTY WINDOW))
             (NEWVALUE (INSPECTW.FETCH WINDOW PROPERTY))
             ITEMSELECTED? NEWVALUEREGION)
            (OR DATUM (ERROR WINDOW " doesn't have a DATUM"))
            (OR OLDVALUEITEM (ERROR "No value for a property in an INSPECTW" WINDOW))
                                                        ; if value being replace is selected, deselect it and reselect it
                                                        ; when finished
            (COND
               ((EQ OLDVALUEITEM (WINDOWPROP WINDOW 'CURRENTITEM))
                (SETQ ITEMSELECTED? T)
                (\ITEMW.DESELECTITEM OLDVALUEITEM WINDOW)))
            (replace ITEMINFO of OLDVALUEITEM with NEWVALUE)    ; erase old stuff
            (DSPFILL (fetch (SELECTABLEITEM SELECTABLEITEMREGION) of OLDVALUEITEM)
                  (DSPTEXTURE NIL WINDOW)
                  'REPLACE WINDOW)
            (PROG1 [SETQ NEWVALUEREGION (replace (SELECTABLEITEM SELECTABLEITEMREGION) of OLDVALUEITEM
                                           with (PRINTATBOX NEWVALUE WINDOW (fetch (SELECTABLEITEM
                                                                                     SELECTABLEITEMREGION
                                                                                     )
                                                                                of OLDVALUEITEM]
                (EXTENDEXTENT WINDOW NEWVALUEREGION)
                (COND
                   (ITEMSELECTED? (\ITEMW.SELECTITEM OLDVALUEITEM WINDOW))))])
```

(INSPECTW.FETCH

```
[LAMBDA (INSPECTW PROPERTY)                             ; Edited 3-Feb-87 16:51 by jop
```

;; retrieves the property value from an inspect window

```
(APPLY* (OR (WINDOWPROP INSPECTW 'FETCHFN)
            (ERROR INSPECTW " doesn't have a FETCHFN"))
        (OR (WINDOWPROP INSPECTW 'DATUM)
            (ERROR INSPECTW " doesn't have a DATUM"))
        PROPERTY])
```

(INSPECTW.PROPERTIES

```
[LAMBDA (INSPECTW)                                      ; Edited 3-Feb-87 16:52 by jop
```

;; gets the list of properties from an INSPECTW.

```
(PROG [(PROPERTIES (WINDOWPROP INSPECTW 'PROPERTIES]
      (RETURN (COND
                 ((OR (NULL PROPERTIES)
                      (LISTP PROPERTIES))
                  PROPERTIES)
                 (T                                     ; allow PROPERTIES to be a function
                    (APPLY* PROPERTIES (WINDOWPROP INSPECTW 'DATUM])
```

(DECODE.WINDOW.ARG

```
[LAMBDA (WHERESPEC WIDTH HEIGHT TITLE BORDER NOOPENFLG)         ; Edited 3-Feb-87 16:48 by jop
```

;; standard useful routine for decoding a window specification arg. WHERESPEC can be a window, a region, a position or NIL. If WHERESPEC is ;; a window, the other args are ignored. This allows programs to override defaults by explicitly providing a window. If a position or NIL, WIDTH and ;; HEIGHT are the dimensions of the new window. The returned window will be entirely on the screen, dimensions permitting.

```
(COND
   ((WINDOWP WHERESPEC)
    WHERESPEC)
   (T (CREATEW (COND
```

```
                    ((REGIONP WHERESPEC)
                     (MAKEWITHINREGION WHERESPEC))
                    [(AND (NUMBERP WIDTH)
                          (NUMBERP HEIGHT))
                     (COND
                        [(POSITIONP WHERESPEC)
                         (MAKEWITHINREGION (CREATEREGION (fetch (POSITION XCOORD) of WHERESPEC)
                                                         (fetch (POSITION YCOORD) of WHERESPEC)
                                                         (WIDTHIFWINDOW WIDTH BORDER)
                                                         (HEIGHTIFWINDOW HEIGHT TITLE BORDER]
                        (T (GETBOXREGION (WIDTHIFWINDOW WIDTH BORDER)
                                         (HEIGHTIFWINDOW HEIGHT TITLE BORDER)
                                         NIL NIL NIL (CONCAT "Specify position for " TITLE]
                    (T NIL))
                TITLE BORDER NOOPENFLG])
```

(DEFAULT.INSPECTW.PROPCOMMANDFN

```
[LAMBDA (PROPERTY DATUM INSPECTW)                          ; Edited  1-Dec-96 20:16 by rmk:
                                                           ; Edited 22-Jun-87 16:41 by jop
```

;; allows the user to select a menu item to change the property in an inspect window.

```
(SELECTQ [MENU (COND
                    ((type? MENU SetPropertyMenu)
                     SetPropertyMenu)
                    (T (SETQ SetPropertyMenu (create MENU
                                                     ITEMS _ '((Set 'SET "Allows a new value to be entered")
                                                               (Inspect 'INSPECT]
     (SET [ERSETQ (PROG ((OLDVALUEITEM (ITEMOFPROPERTYVALUE PROPERTY INSPECTW))
                         (PWINDOW (GETPROMPTWINDOW INSPECTW 3))
                         NEWVALUE)
                        (RESETLST
                            (RESETSAVE (\ITEMW.FLIPITEM OLDVALUEITEM INSPECTW)
                                   (LIST '\ITEMW.FLIPITEM OLDVALUEITEM INSPECTW))
                            (RESETSAVE (TTYDISPLAYSTREAM PWINDOW))
                            (RESETSAVE (TTY.PROCESS (THIS.PROCESS)))
                            (CLEARBUF T T)
                            (printout T "Eval> ")
                            (SETQ NEWVALUE (CL:FUNCALL XCL:*EVAL-FUNCTION* (LISPXREAD T T)))
                                                           ; clear tty buffer because it sometimes has stuff left.
                            (CLEARBUF T T))
                        (REMOVEPROMPTWINDOW INSPECTW)
                        (RETURN (INSPECTW.REPLACE INSPECTW PROPERTY NEWVALUE])
     (INSPECT (INSPECT PROPERTY))
     NIL])
```

(DEFAULT.INSPECTW.VALUECOMMANDFN

```
[LAMBDA (VALUE PROPERTY DATUM WINDOW)                      ; Edited  2-Sep-2022 18:06 by rmk
                                                           ; Edited 24-Aug-2022 23:38 by rmk
                                                           ; Edited 28-Jan-93 16:50 by jds
```

;; allows the user to choose a way to inspect a value in a window

```
(SELECTQ (TYPENAME VALUE)
     ((LITATOM NEW-ATOM)
          (COND
             (VALUE (INSPECT/ATOM VALUE T))
             (T (printout PROMPTWINDOW T "Can't inspect NIL.")
                (until (MOUSESTATE UP))
                (CLRPROMPT))))
     (BITMAP (INSPECT/BITMAP VALUE))
     ((FIXP SMALLP FLOATP)
          (printout PROMPTWINDOW T "Can't Inspect " VALUE)
          (until (MOUSESTATE UP))
          (CLRPROMPT))
     (LISTP                                                ; find out how to inspect the list.
           (INSPECT/LISTP VALUE))
     (SELECTQ [MENU (COND
                       ((type? MENU InspectMenu)
                        InspectMenu)
                       (T (SETQ InspectMenu (create MENU
                                                    ITEMS _ '((Inspect 'INSPECT/VALUE]
          (INSPECT/VALUE
               (INSPECT VALUE))
          NIL])
```

(DEFAULT.INSPECTW.TITLECOMMANDFN

```
[LAMBDA (INSPECTW DATUM)                                   ; Edited  4-Sep-2022 13:42 by rmk
                                                           ; Edited 25-Aug-2022 00:00 by rmk
                                                           (* rrb "18-Apr-84 17:57")
```

```
   (SELECTQ [MENU (COND
                    ((type? MENU ItemWCommandMenu)
                     ItemWCommandMenu)
                    (T (SETQ ItemWCommandMenu (create MENU
                                                      ITEMS _ '((ReFetch 'REFETCH "ReFetches and redisplays the
                                                                                  object's fields")
```

```
                                                                (IT_datum
                                                                 'SETIT "sets the variable IT to the object
                                                                 inspected in this window.")
                                                                (IT_selection
                                                                 'SETITTOSEL "sets the variable IT to the item
                                                                 selected in this window.")
                                                                ("Add  tag" 'ADDTAG "add a mnemonic tag to this
                                                                       window's title"]
        (REFETCH (INSPECTW.REDISPLAY INSPECTW))
        (SETIT (SETQ IT DATUM))
        (SETITTOSEL (COND
                       [(WINDOWPROP INSPECTW 'CURRENTITEM)
                        (SETQ IT (fetch (SELECTABLEITEM ITEMINFO) of (WINDOWPROP INSPECTW 'CURRENTITEM]
                       (T (PROMPTPRINT "No item has been selected from this window."))))
        (ADDTAG [LET (POS TAG (OLDTAG (WINDOWPROP INSPECTW 'INSPECTTAG))
                      (TITLE (WINDOWPROP INSPECTW 'TITLE))
                      (PWINDOW (GETPROMPTWINDOW INSPECTW 1)))
                     (RESETLST
                         (RESETSAVE (TTYDISPLAYSTREAM PWINDOW))
                         (RESETSAVE (TTY.PROCESS (THIS.PROCESS)))
                         (CLEARBUF T T)
                         (PRINTOUT T "Tag> ")
                         (SETQ TAG (CL:READ-LINE T))
                         (CLEARBUF T T)
                         (REMOVEPROMPTWINDOW INSPECTW)
                         (SETQ TAG (CL:IF (EQ 0 (NCHARS TAG))
                                          NIL
                                          (CONCAT TAG ": ")))
                         (WINDOWPROP INSPECTW 'INSPECTTAG TAG)  ; Remove the old tag, if any
                         (CL:WHEN (AND OLDTAG (SETQ POS (STRPOS OLDTAG TITLE 1 NIL T T)))
                             (SETQ TITLE (SUBSTRING TITLE POS)))
                         (WINDOWPROP INSPECTW 'TITLE (CL:IF TAG
                                                            (CONCAT TAG TITLE)
                                                            TITLE)))])
        NIL])
```

(\SELITEM.FROM.PROPERTY

```
  [LAMBDA (INSPECTW PROPERTY)                                   (* rrb " 6-MAR-82 17:50")
    (for SELITEM in (WINDOWPROP INSPECTW 'SELECTABLEITEMS) when (AND (EQ (fetch (SELECTABLEITEM ITEMINFO)
                                                                            of SELITEM)
                                                                        PROPERTY)
                                                                    (EQ (fetch (SELECTABLEITEM ITEMINFOTYPE)
                                                                           of SELITEM)
                                                                        'PROPERTY))
       do (RETURN SELITEM])
```

(\INSPECT.COMPUTE.TITLE

```
  [LAMBDA (TITLE DATUM WINDOW)                                  ; Edited  3-Sep-2022 23:46 by rmk
                                                                ; Edited  2-Sep-2022 18:05 by rmk
                                                                ; Edited 24-Aug-2022 23:35 by rmk
                                                                ; Edited 18-Mar-87 15:23 by jrb:
                                                                ; computes the title for an inspectw from its title field and its
                                                                ; datum.
    (LET (VALUE)
         (COND
            ((NULL TITLE)
             (LET ((*PRINT-LEVEL* 3)
                   (*PRINT-LENGTH* 4))
                  (CL:PRINC-TO-STRING DATUM)))
            ((EQ TITLE 'DON'T)                                  ; no title
             NIL)
            ((LITATOM TITLE)                                    ; it is a function to compute the title.
             (COND
                ((NEQ (SETQ VALUE (APPLY* TITLE DATUM WINDOW))
                      'DON'T)
                 VALUE)
                (T NIL)))
            (T TITLE])
```

(LEVELEDFORM

```
  [LAMBDA (EXP CARLEV CDRLEV)                                   ; Edited  3-Feb-87 16:35 by jop
    ;; returns a copy of EXP that is abbreviated at CARLEV depth in the car direction and CDRLEV depth in the CDR direction
    (COND
       ((NLISTP EXP)
        EXP)
       ((EQ CARLEV 0)
        '&)
       (T (CONS (LEVELEDFORM (CAR EXP)
                             (SUB1 CARLEV)
                             CDRLEV)
                (COND
                   [(EQ CDRLEV 0)
                    (COND
```

```
                    ((CDR EXP)
                     '(--]
                (T (LEVELEDFORM (CDR EXP)
                          CARLEV
                          (SUB1 CDRLEV])
```

(MAKEWITHINREGION

```
[LAMBDA (REGION LIMITREGION)                                    ; Edited 3-Feb-87 16:53 by jop
```

;; moves REGION so that it is entirely on the screen.

```
(DECLARE (GLOBALVARS WHOLEDISPLAY))
(PROG [X (LIMITREGION (COND
                          (LIMITREGION (OR (REGIONP LIMITREGION)
                                           (\ILLEGAL.ARG LIMITREGION)))
                          (T WHOLEDISPLAY]
      [COND
         ((ILESSP (fetch (REGION LEFT) of REGION)
                  (SETQ X (fetch (REGION LEFT) of LIMITREGION)))
          (replace (REGION LEFT) of REGION with X))
         ((IGREATERP (fetch (REGION PRIGHT) of REGION)
                  (SETQ X (fetch (REGION PRIGHT) of LIMITREGION)))
          (replace (REGION LEFT) of REGION with (IMAX 0 (IDIFFERENCE (SUB1 X)
                                                                     (fetch (REGION WIDTH) of REGION]
      [COND
         ((ILESSP (fetch (REGION BOTTOM) of REGION)
                  (SETQ X (fetch (REGION BOTTOM) of LIMITREGION)))
          (replace (REGION BOTTOM) of REGION with X))
         ((IGREATERP (fetch (REGION PTOP) of REGION)
                  (SETQ X (fetch (REGION PTOP) of LIMITREGION)))
          (replace (REGION BOTTOM) of REGION with (IMAX 0 (IDIFFERENCE (SUB1 X)
                                                                       (fetch (REGION HEIGHT) of REGION]
      (RETURN REGION])

)

(DEFINEQ
```

(ITEMW.REPAINTFN

```
[LAMBDA (WINDOW REGION)                                         ; Edited 3-Feb-87 16:31 by jop
                                                                ; repaints the selectable items in a window.
   [for SELITEM in (WINDOWPROP WINDOW 'SELECTABLEITEMS) bind SELECTABLEITEMREGION
      do (COND
            ((REGIONSINTERSECTP REGION (SETQ SELECTABLEITEMREGION (fetch (SELECTABLEITEM SELECTABLEITEMREGION)
                                                                    of SELITEM)))
             (PRINTATBOX (fetch (SELECTABLEITEM ITEMINFO) of SELITEM)
                    WINDOW SELECTABLEITEMREGION]                ; if there is a selected item, flip it too in case some of it was in the
                                                                ; newly exposed area.
   (AND (WINDOWPROP WINDOW 'CURRENTITEM)
        (\ITEMW.FLIPITEM (WINDOWPROP WINDOW 'CURRENTITEM)
              WINDOW])
```

(\ITEM.WINDOW.BUTTON.HANDLER

```
[LAMBDA (WINDOW)                                                ; Edited 3-Feb-87 16:45 by jop
```

;; handles button events for item windows. Basically calls left or middle button handler.

```
(COND
   ((LASTMOUSESTATE LEFT)
    (\ITEM.WINDOW.SELECTION.HANDLER WINDOW))
   ((LASTMOUSESTATE MIDDLE)
    (\INSPECTW.COMMAND.HANDLER WINDOW])
```

(\ITEM.WINDOW.SELECTION.HANDLER

```
[LAMBDA (WINDOW)                                                ; Edited 2-Feb-87 17:25 by jop
```

;; selects an ITEM from the window. If there is an item selected already, it is deselected. An ITEM is a list whose CAR is a region.

```
(PROG ((SELECTABLEITEMS (WINDOWPROP WINDOW 'SELECTABLEITEMS))
       NOW PREVIOUS BUTTON OLDPOS REG)
      (COND
         ((NULL SELECTABLEITEMS)                                ; no items, don't do anything.
          (RETURN)))                                            ; note which button is down.
      (COND
         ((LASTMOUSESTATE LEFT)
          (SETQ BUTTON 'LEFT))
         ((LASTMOUSESTATE MIDDLE)
          (SETQ BUTTON 'MIDDLE))
         (T                                                     ; no button down, not interested.
            (RETURN)))
      (TOTOPW WINDOW)
      (SETQ REG (WINDOWPROP WINDOW 'REGION))                    ; note current item selection.
      [SETQ NOW (IN/ITEM? SELECTABLEITEMS (SETQ OLDPOS (CURSORPOSITION NIL WINDOW]
      (SETQ PREVIOUS (WINDOWPROP WINDOW 'CURRENTITEM))
  FLIP
                                                                ; turn off old selection.
      (\ITEMW.DESELECTITEM PREVIOUS WINDOW)
```

```
          (\ITEMW.SELECTITEM (SETQ PREVIOUS NOW)
                 WINDOW)
     LP                                                    ; wait for a button up or move out of region
          (GETMOUSESTATE)
          (COND
             ((NOT (LASTMOUSESTATE (OR LEFT MIDDLE)))      ; button up, return
              (AND NOW (WINDOWPROP WINDOW 'SELECTIONFN)
                   (APPLY* (WINDOWPROP WINDOW 'SELECTIONFN)
                          [COND
                             ((EQ 'PROPERTY (fetch (SELECTABLEITEM ITEMINFOTYPE) of NOW))
                              (fetch (SELECTABLEITEM ITEMINFO) of NOW))
                             (T (CAR (fetch (SELECTABLEITEM ITEMINFOTYPE) of NOW]
                          (NEQ (fetch (SELECTABLEITEM ITEMINFOTYPE) of NOW)
                               'PROPERTY)
                          WINDOW))
              (RETURN))
             ((NOT (INSIDE? REG LASTMOUSEX LASTMOUSEY))    ; outside of region, return
              (\ITEMW.DESELECTITEM PREVIOUS WINDOW)
              (RETURN))
             ([EQ PREVIOUS (SETQ NOW (IN/ITEM? SELECTABLEITEMS (CURSORPOSITION NIL WINDOW OLDPOS]
              (GO LP))
             (T (GO FLIP])
```

(\INSPECTW.COMMAND.HANDLER

```
[LAMBDA (INSPECTW)                                         ; Edited 8-Apr-87 16:40 by jop
```

;; the user has middle buttoned in an ITEM window. Apply the selected item's COMMANDFN to the selected item and the window. Often the ;; commandfn will put up another menu.

```
(WITH-INSPECTOR-ENV (WINDOWPROP INSPECTW 'PROFILE)
       (COND
          [(INSIDEP (DSPCLIPPINGREGION NIL INSPECTW)
                   (LASTMOUSEX INSPECTW)
                   (LASTMOUSEY INSPECTW))                  ; inside of interior
           (PROG ((SELITEM (WINDOWPROP INSPECTW 'CURRENTITEM))
                  COMMANDFN INFO)
                 (RETURN (COND
                            [SELITEM (COND
                                        ((NULL (SETQ COMMANDFN (fetch (SELECTABLEITEM COMMANDFN) of SELITEM))
                                               )           ; special case of NIL command fn
                                         (PROMPTPRINT "There is no change function for this window."))
                                        ((STRINGP COMMANDFN)
                                         (PROMPTPRINT COMMANDFN))
                                        (T
```

;; check to see if the selected item is a property or a value. This distinction is because the value one ;; needs an extra argument. The selected item is considered to be a property if it is one of the ;; properties of the window.

```
                                           (ERSETQ (COND
                                                      ((EQ (SETQ INFO (fetch (SELECTABLEITEM ITEMINFOTYPE)
                                                                             of SELITEM))
                                                           'PROPERTY)
                                                       ; the selected item is a property. Call the command fn in
                                                       ; property form.
                                                       (APPLY* COMMANDFN (fetch (SELECTABLEITEM ITEMINFO)
                                                                                of SELITEM)
                                                              (WINDOWPROP INSPECTW 'DATUM)
                                                              INSPECTW))
                                                      (T
```

;; the selected item is a value Call the command fn in value form. For values, the item info ;; type is a cons whose CAR is the property

```
                                                         (APPLY* COMMANDFN (fetch (SELECTABLEITEM ITEMINFO)
                                                                                  of SELITEM)
                                                                (CAR INFO)
                                                                (WINDOWPROP INSPECTW 'DATUM)
                                                                INSPECTW]
                            (T (PROMPTPRINT "This is the command button. You must select an item with the
                                      left button before choosing a command.")
                               (until (MOUSESTATE UP))
                               (CLRPROMPT]
          (T                                               ; inside border or title Call the window's TITLECOMMANDFN
             (APPLY* (OR (WINDOWPROP INSPECTW 'TITLECOMMANDFN)
                         (FUNCTION DEFAULT.INSPECTW.TITLECOMMANDFN))
                     INSPECTW
                     (WINDOWPROP INSPECTW 'DATUM])
```

(\ITEM.WINDOW.SET.STACK.ARG

```
[LAMBDA (VARNAME FRAME WINDOW)                             ; Edited 3-Feb-87 16:52 by jop
```

;; the PropCommandFn for itemw windows onto stack frames.

```
(SELECTQ [MENU (COND
                  ((type? MENU SetStackMenu)
                   SetStackMenu)
                  (T (SETQ SetStackMenu (create MENU
                                                ITEMS _ '((Set 'SET "Changes the value of this stack
```

```
                                                                variable"]
        (SET (OR (STACKP FRAME)
                 (\ILLEGAL.ARG FRAME))
             [ERSETQ (PROG ((OLDVALUEITEM (ITEMOFPROPERTYVALUE VARNAME WINDOW))
                            NEWVALUE)                                    ; decode the argument position
                    ;; insist that the arg being set has a real name.  following is the code to allow any var to be set: (SETQ ARGN (COND
                    ;; ((FRAMESCAN VARNAME FRAME)) ((STRPOS VARNAME '*arg' 1 T) (COND ((SMALLP (SUBATOM VARNAME 5
                    ;; -1))) (T (PROMPTPRINT 'Can't set that arg.') (RETURN)))) ((STRPOS VARNAME '*prg' 1 T) (COND ((SETQ ARGN
                    ;; (SMALLP (SUBATOM VARNAME 5 -1))) (IPLUS ARGN (STKNARGS FRAME))) (T (PROMPTPRINT 'Can't set that
                    ;; arg.') (RETURN))))))
                          (COND
                             ((FRAMESCAN VARNAME FRAME))
                             (T (PROMPTPRINT "Can't set that arg.")
                                (RETURN)))
                          (RESETLST
                              (RESETSAVE (\ITEMW.FLIPITEM OLDVALUEITEM WINDOW)
                                     (LIST '\ITEMW.FLIPITEM OLDVALUEITEM WINDOW))
                              (RESETSAVE (TTY.PROCESS (THIS.PROCESS)))
                              (CLRPROMPT)
                              (printout T "Enter the new value for " VARNAME "." T "The expression read will
                                     be EVALuated." T "> ")
                              (SETQ NEWVALUE (EVAL (READ T T))))
                          (RETURN (INSPECTW.REPLACE WINDOW VARNAME NEWVALUE])
        NIL])
```

(REPLACESTKARG

```
[LAMBDA (FRAMESPEC WHICHSPEC NEWVALUE)                               ; Edited  3-Feb-87 16:54 by jop
```

;; StoreFn for the ITEMW that inspects back trace frames.

```
(COND
   ((NULL (CDR WHICHSPEC))                                           ; this is a dummy which is a function name.  it has no value
    NIL)
   ((LISTP FRAMESPEC)
    (REPLACESTKARG (CAR (NTH FRAMESPEC (CAR WHICHSPEC)))
           (CDR WHICHSPEC)
           NEWVALUE))
   (T (PROG NIL
            (OR (STACKP FRAMESPEC)
                (\ILLEGAL.ARG FRAMESPEC))
            (RETURN (SETSTKARG (COND
                                  ((LISTP WHICHSPEC)                 ; CAR is name, CADR is offset
                                   (CADR WHICHSPEC))
                                  ((FRAMESCAN WHICHSPEC FRAMESPEC))
                                  (T (PROMPTPRINT "Can't set that arg.")
                                     (RETURN)))
                               FRAMESPEC NEWVALUE])
```

(IN/ITEM?

```
[LAMBDA (ITEMS POS)                                                  (* rrb "28-AUG-83 12:18")
  (PROG ((XPOS (fetch XCOORD of POS))
         (YPOS (fetch YCOORD of POS)))
        (RETURN (for ITEM in ITEMS when (AND (fetch (SELECTABLEITEM SELECTABLEITEMREGION) of ITEM)
                                             (INSIDE? (fetch (SELECTABLEITEM SELECTABLEITEMREGION) of ITEM)
                                                    XPOS YPOS))
                   do (RETURN ITEM])
```

(\ITEMW.DESELECTITEM

```
[LAMBDA (ITEM WINDOW)                                                ; Edited  3-Feb-87 15:46 by jop
```

;; deselects ITEM from window

```
(AND ITEM (\ITEMW.FLIPITEM ITEM WINDOW))
(WINDOWPROP WINDOW 'CURRENTITEM NIL])
```

(\ITEMW.SELECTITEM

```
[LAMBDA (ITEM WINDOW)                                                ; Edited  3-Feb-87 15:46 by jop
```

;; selects an ITEM in WINDOW

```
(AND ITEM (\ITEMW.FLIPITEM ITEM WINDOW))
(WINDOWPROP WINDOW 'CURRENTITEM ITEM])
```

(\ITEMW.CLEARSELECTION

```
[LAMBDA (INSPECTW)                                                   ; Edited  3-Feb-87 16:56 by jop
```

;; clears the selection from an inspect window

```
(PROG [(CURRENTITEM (WINDOWPROP INSPECTW 'CURRENTITEM]
      (AND CURRENTITEM (\ITEMW.DESELECTITEM CURRENTITEM INSPECTW))
      (RETURN INSPECTW])
```

(\ITEMW.FLIPITEM

```
[LAMBDA (ITEM DS)                                                    ; Edited  3-Feb-87 15:46 by jop
```

```
    ;; flips the region of an item
    (LET ((REG (fetch (SELECTABLEITEM SELECTABLEITEMREGION) of ITEM)))
         (BLTSHADE BLACKSHADE DS (fetch LEFT of REG)
                (fetch BOTTOM of REG)
                (fetch WIDTH of REG)
                (fetch HEIGHT of REG)
                'INVERT])
```

(PRINTANDBOX

```
  [LAMBDA (EXP STREAM LFTMARGIN MINSPACE)                    ; Edited  4-May-87 14:35 by jop
    ;; prints EXP on WINDOW starting at LFTMARGIN and returns the box taken by the characters.  Leaves at least MINSPACE points.
    ;; set the left margin so that at least nothing will CR past it.  This does not handle multiple line values.
    (PROG ((STRM (\OUTSTREAMARG STREAM))
           PREVRM PREVLM YSTART YEND HGHT)
          (SETQ PREVRM (DSPRIGHTMARGIN 50000 STRM))          ; so that it won't auto carrage return.
          (SETQ PREVLM (DSPLEFTMARGIN LFTMARGIN STRM))
          (AND (FIXP MINSPACE)
               (RELMOVETO MINSPACE 0 STRM))
          (COND
             ((IGREATERP (DSPXPOSITION NIL STRM)
                    LFTMARGIN)
              (TERPRI STRM)))
          (DSPXPOSITION LFTMARGIN STRM)
          (SETQ YSTART (DSPYPOSITION NIL STRM))
          (RETURN (PROG1 [create REGION
                                 LEFT _ LFTMARGIN
                                 BOTTOM _ [PROGN (CL:PRIN1 EXP STRM)
                                                 (IDIFFERENCE (SETQ YEND (DSPYPOSITION NIL STRM))
                                                        (FONTPROP STRM 'DESCENT]
                                 HEIGHT _ (IPLUS (SETQ HGHT (IDIFFERENCE YSTART YEND))
                                                 (FONTPROP STRM 'HEIGHT))
                                 WIDTH _ (COND
                                            ((IGREATERP HGHT 0)   ; printing the thing did an overflow;  use at least the width of the
                                                                  ; window.
                                             (IMAX (IDIFFERENCE (DSPXPOSITION NIL STRM)
                                                                LFTMARGIN)
                                                   (IDIFFERENCE (fetch (REGION WIDTH) of (DSPCLIPPINGREGION NIL
                                                                                                          STRM))
                                                                LFTMARGIN)))
                                            (T (IDIFFERENCE (DSPXPOSITION NIL STRM)
                                                            LFTMARGIN]
                           (DSPRIGHTMARGIN PREVRM STRM)
                           (DSPLEFTMARGIN PREVLM STRM))])
```

(PRINTATBOX

```
  [LAMBDA (EXP WINDOW OLDBOX)                                ; Edited  3-Feb-87 16:31 by jop
    ;; prints EXP in place of what used to be in oldbox and returns the new box.
    (DSPFILL OLDBOX NIL 'REPLACE WINDOW)
    (MOVETO (fetch LEFT of OLDBOX)
            (IDIFFERENCE (fetch PTOP of OLDBOX)
                   (FONTPROP (DSPFONT NIL WINDOW)
                          'ASCENT))
            WINDOW)
    (PRINTANDBOX EXP WINDOW (fetch LEFT of OLDBOX])
```

(ITEMOFPROPERTYVALUE

```
  [LAMBDA (PROPERTY WINDOW)                                  ; Edited  3-Feb-87 16:53 by jop
    ;; returns the selectableitem structure that corresponds to the value of a property in an inspectw.  Knows the way INSPECTW are created.
    (CADR (MEMB (\SELITEM.FROM.PROPERTY WINDOW PROPERTY)
                (WINDOWPROP WINDOW 'SELECTABLEITEMS])

)

(DEFINEQ
```

(\ITEM.WINDOW.COPY.HANDLER

```
  [LAMBDA (WINDOW)                                           ; Edited  4-Sep-2022 08:58 by rmk
                                                             ; Edited  2-Feb-87 17:27 by jop
    ;; copy selects an ITEM from the window.  An ITEM is an instance of record SELECTABLEITEM.
    (PROG ((SELECTABLEITEMS (WINDOWPROP WINDOW 'SELECTABLEITEMS))
           CURRENTITEM SMASHPOS NEWITEM)
          (COND
             ((NULL SELECTABLEITEMS)                         ; no items, don't do anything.
              (RETURN)))
      LP  (TOTOPW WINDOW)                                    ; note current item selection.
          [SETQ NEWITEM (IN/ITEM? SELECTABLEITEMS (SETQ SMASHPOS (CURSORPOSITION NIL WINDOW]
          [COND
             ((NEQ CURRENTITEM NEWITEM)
              (COND
```

```
                (CURRENTITEM                                        ; turn off old selection.
                       (\ITEMW.FLIPCOPY CURRENTITEM WINDOW)))
             (COND
                ((SETQ CURRENTITEM NEWITEM)
                 (\ITEMW.FLIPCOPY CURRENTITEM WINDOW]               ; wait for a button up or move out of region
      LP2 (BLOCK)
          (COND
             ((NOT (.COPYKEYDOWNP.))                                ; Finished, copy selected item
              [COND
                 (CURRENTITEM (\ITEMW.FLIPCOPY CURRENTITEM WINDOW)
                        (BKSYSBUF.GENERAL (fetch (SELECTABLEITEM ITEMINFO) of CURRENTITEM)))
                 (T (BKSYSBUF.GENERAL (WINDOWPROP WINDOW 'DATUM]
              (RETURN))
             ((MOUSESTATE UP)                                       ; button up, no action
              (GO LP2))
             (T (GO LP])
```

(\ITEMW.FLIPCOPY

```
  [LAMBDA (ITEM DS)                                                 ; Edited  3-Feb-87 16:56 by jop
    ;; flips the copy selection region of an item
    (LET ((REG (fetch (SELECTABLEITEM SELECTABLEITEMREGION) of ITEM)))
         (BLTSHADE GRAYSHADE DS (fetch LEFT of REG)
                (fetch BOTTOM of REG)
                (fetch WIDTH of REG)
                2
                'INVERT])
```

(BKSYSBUF.GENERAL

```
  [LAMBDA (OBJECT)                                                  ; Edited 10-Jul-91 13:25 by jds
    ;; Does a slightly more intelligent BKSYSBUF than just stuffing the print name as characters
    (LET ((TYPE (TYPENAME OBJECT)))
         (SELECTQ TYPE
             ((STRINGP LITATOM NEW-ATOM)
                 (BKSYSBUF OBJECT T))
             (LISTP (bind (SEPR _ '%() do (BKSYSBUF SEPR)
                                          (SETQ SEPR '% )
                                          (BKSYSBUF.GENERAL (CAR OBJECT))
                    repeatuntil (NLISTP (SETQ OBJECT (CDR OBJECT))) finally (COND
                                                                               (OBJECT
                                                                                ; Dotted cdr
                                                                                       (BKSYSBUF " . ")
                                                                                       (BKSYSBUF.GENERAL OBJECT)
                                                                                       ))
                                                                          (BKSYSBUF '%))))
             (COND
                ((NUMBERP OBJECT)
                 (BKSYSBUF OBJECT))
                (T (RESETVARS ((PRXFLG T))
                              (LET ((*PRINT-BASE* 8)
                                    (*PRINT-RADIX* T))
                                   (BKSYSBUF (LIST '\VAG2 (\HILOC OBJECT)
                                                          (\LOLOC OBJECT))
                                         T])

)

(DECLARE%: EVAL@COMPILE

(RECORD SELECTABLEITEM (SELECTABLEITEMREGION COMMANDFN ITEMINFO ITEMINFOTYPE))
)

(RPAQQ MAXINSPECTARRAYLEVEL 300)

(RPAQQ MAXINSPECTCDRLEVEL 50)

(RPAQQ MinSpaceBetweenProperyAndValue 8)

(RPAQQ MaxInspectorPropertyValueWidth 250)

(RPAQQ MaxValueLeftMargin 250)

(RPAQQ PropertyLeftMargin 2)

;; functions for the inspector

(DEFINEQ
```

(INSPECT

```
  [LAMBDA (ITEM ASTYPE WHERE TAG)
    ;; Edited 17-Sep-2022 22:30 by rmk
    ;; Edited 12-Sep-2022 20:54 by rmk
```

```
;; Edited  3-Sep-2022 23:32 by rmk
;; Edited  2-Sep-2022 17:59 by rmk
;; Edited 24-Aug-2022 23:32 by rmk:  Added optional TAG as a mnemonic packed onto window title
;; Edited  1-Dec-96 21:09 by rmk:
;; Edited  2-Feb-87 17:09 by jop
;; sets up a window that allows inspection.
(DECLARE (SPECVARS WHERE TAG))
(LET ((ITEMTYPE (TYPENAME ITEM))
      IWINDOW INSPECTINFO)
     (CL:SETQ IWINDOW (COND
                          (ASTYPE                                   ; if ASTYPE is given, always inspect it as that type.  This
                                                                    ; provides a way of overriding macros.
                                 (INSPECT/DATATYPE ITEM ASTYPE WHERE TAG))
                          [(SETQ INSPECTINFO (for IMACRO in INSPECTMACROS
                                                when (COND
                                                        [(LISTP (CAR IMACRO))
                                                         (COND
                                                            ((EQ (CAAR IMACRO)
                                                                 'FUNCTION)
                                                             (APPLY* (CADAR IMACRO)
                                                                     ITEM))
                                                            (T (ERROR "ERROR in INSPECTMACROS specification"
                                                                      IMACRO]
                                                        (T (EQ (CAR IMACRO)
                                                               ITEMTYPE)))
                                                do (RETURN IMACRO)))
                           (COND
                              ((LISTP (CDR INSPECTINFO))            ; inspect information is a list of arguments to
                                                                    ; INSPECTW.CREATE
                               (\APPLYINSPECTMACRO ITEM (CDR INSPECTINFO)
                                      WHERE TAG))
                              (T                                    ; if inspect information is an atom, apply it to the ITEM.
                                 (APPLY* (CDR INSPECTINFO)
                                         ITEM
                                         (CAR INSPECTINFO)
                                         WHERE TAG]
                          [ITEM (SELECTQ ITEMTYPE
                                    (LITATOM (INSPECT/ATOM ITEM NIL WHERE TAG))
                                    (LISTP                          ; find out how to inspect the list.
                                           (INSPECT/LISTP ITEM WHERE TAG))
                                    (ARRAYP (INSPECT/ARRAY ITEM NIL WHERE TAG))
                                    (HARRAYP (INSPECT/HARRAYP ITEM WHERE TAG))
                                    (BITMAP (INSPECT/BITMAP ITEM WHERE TAG))
                                    (CCODEP (INSPECTCODE ITEM WHERE TAG))
                                    (NIL (INSPECT/TYPELESS ITEM WHERE TAG))
                                    (LET [(DTD (\GETDTD (NTYPX ITEM]
                                         (COND
                                            ((fetch DTDHUNKP of DTD)
                                             (INSPECT/HUNK ITEM WHERE (fetch DTDGCTYPE of DTD)
                                                    (fetch DTDSIZE of DTD)
                                                    TAG))
                                            (T (INSPECT/DATATYPE ITEM NIL WHERE TAG]
                          (T (printout PROMPTWINDOW T "Can't Inspect NIL.")
                             NIL)))
     (CL:WHEN (WINDOWP IWINDOW)                                     ; Mark it as an inspect window, so that utilities such as
                                                                    ; WDWHACKS can recognize it
         (WINDOWPROP IWINDOW 'INSPECTWINDOW T))
     IWINDOW])
```

(\APPLYINSPECTMACRO

```
[LAMBDA (DATUM ARGLST WHERE TAG)                                    ; Edited 12-Sep-2022 20:50 by rmk
                                                                    ; Edited  3-Feb-87 15:18 by jop
;; function that calls INSPECTW.CREATE when given the inspect macro information.  Separate because of difficulty of interpreting WHERE
;; argument.
(PROG ((ARGS ARGLST))
      (RETURN (INSPECTW.CREATE DATUM (pop ARGS)
                               (pop ARGS)
                               (pop ARGS)
                               (pop ARGS)
                               (pop ARGS)
                               (pop ARGS)
                               (pop ARGS)
                               (COND
                                  (ARGS                             ; WHERE argument must be evaluated.
                                        (EVAL ARGS))
                                  (T WHERE))
                               (pop ARGS)
                              TAG])
```

(INSPECT/BITMAP

```
[LAMBDA (BITMAP WHERE TAG)                                  ; Edited 12-Sep-2022 20:52 by rmk
                                                            ; Edited  2-Feb-87 17:07 by jop

    ;; asks whether to use the bitmap editor or not

    (SELECTQ [MENU (COND
                       ((type? MENU InspectBitmapMenu)
                        InspectBitmapMenu)
                       (T (SETQ InspectBitmapMenu (create MENU
                                                          ITEMS _ '((fields 'FIELDS "Inspects the fields of the
                                                                    bitmap")
                                                                   (contents 'CONTENTS "Edits the contents of the
                                                                    bitmap."]
          (FIELDS (INSPECT/DATATYPE BITMAP 'BITMAP WHERE TAG))
          (CONTENTS (EVAL.AS.PROCESS (LIST 'EDITBM BITMAP)))
          NIL])
```

(INSPECT/DATATYPE

```
[LAMBDA (DATUM TYPE WHERE TAG)                              ; Edited  1-Jun-2023 22:33 by rmk
                                                            ; Edited 12-Sep-2022 20:58 by rmk
                                                            ; Edited  9-Aug-2022 08:56 by rmk
                                                            ; Edited  1-Dec-96 20:15 by rmk:
                                                            ; Edited  7-Aug-87 10:21 by jop

    ;; creates an inspector window for datatype or record instance DATUM

    (LET (SYSREC DEC)
         (COND
            [(AND TYPE (SETQ DEC (RECLOOK TYPE]
            ((AND TYPE (SETQ DEC (SYSRECLOOK1 TYPE)))
             (SETQ SYSREC T))
            ((SETQ DEC (FINDRECDECL DATUM)))
            ((SETQ DEC (FINDSYSRECDECL DATUM))
             (SETQ SYSREC T)))
         (COND
            (DEC                                            ; The fetchfn and storefn would be more attractive if we had
                                                            ; lexical closures
                 (INSPECTW.CREATE DATUM (INSPECTABLEFIELDNAMES DEC (NULL INSPECTALLFIELDSFLG))
                        `[LAMBDA (INSTANCE FIELD)
                           (RECORDACCESS FIELD INSTANCE ',DEC]
                        [if SYSREC
                            then `[LAMBDA (INSTANCE FIELD NEWVALUE)
                                     (AND (CONFIRM-SET)
                                          (RECORDACCESS FIELD INSTANCE ',DEC '/REPLACE NEWVALUE]
                          else `(LAMBDA (INSTANCE FIELD NEWVALUE)
                                   (RECORDACCESS FIELD INSTANCE ',DEC '/REPLACE NEWVALUE]
                        NIL NIL (if (EQ (CAR DEC)
                                        'BLOCKRECORD)
                                    then
                                    ;; To this by hand  to avoid being fooled by invalid lisp pointers
                                         (CL:FORMAT NIL "<~a @ ~o,~o>" TYPE (\HILOC DATUM)
                                                (\LOLOC DATUM)))
                        NIL NIL WHERE NIL TAG))
            ([SETQ DEC (fetch DTDDESCRS of (\GETDTD (NTYPX DATUM]   ; No user-level declaration, but we can at least fetch raw fields
                                                            ; out of it
             (INSPECTW.CREATE DATUM (for I to (LENGTH DEC) collect I)
                    `[LAMBDA (FIELD INSTANCE)
                       (\INSPECT.DATATYPE.RAW.FETCH FIELD INSTANCE ',DEC]
                    NIL "System datatype: Cann't set any fields" NIL NIL NIL NIL WHERE NIL TAG))
            ((AND (LISTP DATUM)
                  (SELECTQ TYPE
                      (ALIST (CL:WHEN (ALISTP DATUM)
                                      (INSPECT/ALIST DATUM WHERE TAG)
                                      T)
                             (ALISTP DATUM))
                      (PLIST (CL:WHEN (PROPLISTP DATUM)
                                      (INSPECT/PLIST DATUM WHERE TAG)
                                      T))
                      (LIST (INSPECT/TOP/LEVEL/LIST DATUM WHERE TAG))
                      NIL)))
            (T (printout PROMPTWINDOW T "No declaration for " DATUM T "Can not inspect.")
               NIL])
```

(INSPECTABLEFIELDNAMES

```
[LAMBDA (DECL TOPONLYFLG)                                   ; Edited 14-Sep-2023 23:28 by rmk
                                                            ; Edited 15-Jun-2023 15:36 by rmk
                                                            ; Edited  2-Jun-2023 20:18 by rmk
                                                            ; Edited  3-Feb-87 16:51 by jop

    ;; returns the list of record field names suitable for inspecting.  This is everything unless TOPONLYFLG is T which is the case for system records.

    (LET (FIELDNAMES)
         [SETQ FIELDNAMES (COND
                             (TOPONLYFLG (for FIELDNAME in (CDR (RECORDFIELDNAMES DECL T))
                                            when (AND FIELDNAME (NLISTP FIELDNAME)) collect FIELDNAME))
                             (T (REMOVEDUPS (RECORDFIELDNAMES DECL]
         (CL:IF (OR (NEQ 'DATATYPE (CAR (LISTP DECL)))
                    (EQ T INSPECTDONTSORTFIELDS)
```

```
                  (MEMB (CADR DECL)
                        INSPECTDONTSORTFIELDS)
                  (ILEQ (LENGTH FIELDNAMES)
                        5))
            FIELDNAMES
            (SORT FIELDNAMES))])
```

(REMOVEDUPS

```
[LAMBDA (LST)                                          ; Edited  3-Feb-87 16:54 by jop
```

;; removes the duplicate entries from LST.

```
(INTERSECTION LST LST])
```

(INSPECT/ARRAY

```
[LAMBDA (ARRAY BEGINOFFSET WHERE TAG)                  ; Edited 12-Sep-2022 20:55 by rmk
                                                       ; Edited  2-Feb-87 17:06 by jop
```

;; inspects an array

```
(COND
   [(ARRAYP ARRAY)
    (PROG [(FIRSTELT (OR (NUMBERP BEGINOFFSET)
                         (ARRAYORIG ARRAY]
          (RETURN (INSPECTW.CREATE ARRAY (for I from FIRSTELT to (SUB1 (IMIN (IPLUS (ARRAYORIG ARRAY)
                                                                                    (ARRAYSIZE ARRAY))
                                                                               (IPLUS FIRSTELT
                                                                                      MAXINSPECTARRAYLEVEL)))
                                                 collect I)
                                 (FUNCTION ELT)
                                 (FUNCTION /SETA)
                                 NIL NIL NIL NIL NIL WHERE NIL TAG]
   (T (printout PROMPTWINDOW T ARRAY " not an array")
      NIL])
```

(INSPECT/TOP/LEVEL/LIST

```
[LAMBDA (LST WHERE TAG)                                ; Edited 12-Sep-2022 20:56 by rmk
                                                       ; Edited  9-Sep-2022 21:49 by rmk
                                                       ; Edited  2-Feb-87 17:02 by jop
```

;; inspects one level of a list structure via numbered fields.

```
(COND
   ((LISTP LST)
    (INSPECTW.CREATE LST [for I from 1 to MAXINSPECTCDRLEVEL as X on LST collect I
                             finally (COND
                                        (X (NCONC1 $$VAL (COND
                                                           ((NLISTP X)
                                                            '|...|)
                                                           (T '&&]
          (FUNCTION NTHTOPLEVELELT)
          (FUNCTION SETNTHTOPLEVELELT)
          NIL NIL NIL NIL NIL WHERE NIL TAG))
   (T (printout PROMPTWINDOW T LST " not a LISTP")
      NIL])
```

(INSPECT/PROPLIST

```
[LAMBDA (ATOM ALLPROPSFLG WHERE TAG)                   ; Edited 12-Sep-2022 20:59 by rmk
                                                       ; Edited  3-Feb-87 16:51 by jop
```

;; opens an inspect window onto the properties of ATOM

```
(PROG [(PROPS (COND
                 (ALLPROPSFLG (PROPNAMES ATOM))
                 (T (NONSYSPROPNAMES ATOM]
      (RETURN (COND
                 (PROPS (INSPECTW.CREATE ATOM (COND
                                                 (ALLPROPSFLG (FUNCTION PROPNAMES))
                                                 (T (FUNCTION NONSYSPROPNAMES)))
                                         (FUNCTION GETPROP)
                                         (FUNCTION /PUTPROP)
                                         NIL NIL NIL NIL NIL WHERE NIL TAG))
                 (T (PROMPTPRINT (COND
                                    (ALLPROPSFLG "No properties")
                                    (T "No non-system properties")))
                    NIL])
```

(NONSYSPROPNAMES

```
[LAMBDA (ATM)                                          ; Edited  3-Feb-87 16:53 by jop
```

;; returns the properties an atom has that are not SYSPROPS

```
(for PROP in (PROPNAMES ATM) when (NOT (FMEMB PROP SYSPROPS)) collect PROP])
```

(INSPECT/LISTP

```
[LAMBDA (LST WHERE TAG)                                ; Edited 12-Sep-2022 20:49 by rmk
```

; Edited 2-Feb-87 17:05 by jop

;; asks how the user wants to inspect a list and calls the appropriate function.

```
(APPLY* (OR (SELECT.LIST.INSPECTOR LST)
            (FUNCTION NILL))
       LST WHERE TAG])
```

(ALISTP

```
[LAMBDA (LST)
```

; Edited 3-Feb-87 16:48 by jop

;; is LST in alist format?

```
(for ELT in LST always (LISTP ELT])
```

(PROPLISTP

```
[LAMBDA (LST)
```

; Edited 3-Feb-87 16:54 by jop

;; is lst a property list format? Assumes that property names are litatoms.

```
(AND LST (PROG ((LSTPTR LST))
           LP  (COND
                  ((NULL LSTPTR)
                   (RETURN T))
                  ((NLISTP LSTPTR)
                   (RETURN NIL))
                  ((AND (LITATOM (CAR LSTPTR))
                        (LISTP (CDR LSTPTR)))
                   (SETQ LSTPTR (CDDR LSTPTR))
                   (GO LP))
                  (T (RETURN NIL])
```

(INSPECT/ALIST

```
[LAMBDA (ALST WHERE TAG)
```

; Edited 12-Sep-2022 20:59 by rmk
; Edited 2-Feb-87 17:04 by jop

;; opens an inspect window onto an ALIST.

```
(INSPECTW.CREATE ALST (for X in ALST collect (CAR X))
       (FUNCTION ASSOCGET)
       (FUNCTION /ASSOCPUT)
       NIL NIL NIL NIL NIL WHERE NIL TAG])
```

(ASSOCGET

```
[LAMBDA (ALST KEY)
```

; Edited 2-Feb-87 17:04 by jop

;; gets the value associated with a key on an ALST.

```
(CDR (ASSOC KEY ALST])
```

(/ASSOCPUT

```
[LAMBDA (ALST KEY VAL)
```

; Edited 2-Feb-87 17:04 by jop

;; defined to change the order of arguments from what inspector gives to what /PUTASSOC wants.

```
(/PUTASSOC KEY VAL ALST])
```

(INSPECT/PLIST

```
[LAMBDA (PLST WHERE TAG)
```

; Edited 12-Sep-2022 20:57 by rmk
; Edited 2-Feb-87 17:05 by jop

;; opens an inspect window onto an ALIST.

```
(INSPECTW.CREATE PLST (for X in PLST by (CDDR X) collect X)
       (FUNCTION LISTGET)
       (FUNCTION /LISTPUT)
       NIL NIL NIL NIL NIL WHERE NIL TAG])
```

(INSPECT/TYPERECORD

```
[LAMBDA (X WHERE TAG)
```

; Edited 12-Sep-2022 21:05 by rmk
; Edited 2-Feb-87 17:05 by jop

;; inspects X assuming it is a typerecord instance.

```
(INSPECT X (CAR X)
       WHERE TAG])
```

(INSPECT/AS/RECORD

```
[LAMBDA (INSTANCE WHERE TAG)
```

; Edited 12-Sep-2022 21:04 by rmk
; Edited 2-Feb-87 17:03 by jop

;; offers the user a choice of record types to inspect INSTANCE with.

```
(PROG (RECORD)
      (RETURN (AND [SETQ RECORD (MENU (create MENU
                                              ITEMS _ (SORT (for RECDEC in USERRECLST
                                                               when (FMEMB (CAR RECDEC)
                                                                           '(TYPERECORD RECORD))
```

```
                                                        collect (CADR RECDEC)))
                                              WHENHELDFN _ (FUNCTION (LAMBDA (ITEM)
                                                               (PROMPTPRINT "Will inspect the list as
                                                                      if it were an instance of this
                                                                      record type."]
                        (INSPECT INSTANCE RECORD WHERE TAG])
```

(SELECT.LIST.INSPECTOR

```
[LAMBDA (LST)                                                    ; Edited 2-Feb-87 17:05 by jop
```

;; gives the user a choice of how to edit a list.

```
(MENU (create MENU
              ITEMS _ [APPEND '((DisplayEdit 'DEDITE "Edit it with the display editor")
                                (TtyEdit 'STANDARDEDITE "Edit it with the standard editor")
                                (Inspect 'INSPECT/TOP/LEVEL/LIST "Inspect the top level with an inspect
                                         window")
                                ("As a record" 'INSPECT/AS/RECORD "Prompts further for the record type of
                                         this LIST."))
                              [COND
                                 [(ALISTP LST)
                                  '(("As an ALIST" 'INSPECT/ALIST "Inspects the list as a A-List"]
                                 ((PROPLISTP LST)
                                  '(("As a PLIST" 'INSPECT/PLIST "Inspects the list as a property list."]
                              (PROG [(RECDEC (RECLOOK (CAR LST]
                                    (RETURN (COND
                                               ((AND RECDEC (EQ (CAR RECDEC)
                                                                'TYPERECORD))
                                                                ; this is likely to be an instance of the typed record.
                                                (CONS (LIST (CONCAT "As a " (CAR LST))
                                                            ''INSPECT/TYPERECORD
                                                            (CONCAT "Inspects the selected list as an instance
                                                                   of " (CAR LST]
              CENTERFLG _ T])
```

(STANDARDEDITE

```
[LAMBDA (EXPR COMS ATM TYPE IFCHANGEDFN)                         ; Edited 3-Feb-87 16:55 by jop
```

;; version of EDITE that always calls the standard editor.

```
(RESETFORM (EDITMODE 'STANDARD)
       (EDITE EXPR COMS ATM TYPE IFCHANGEDFN])
```

(NTHTOPLEVELELT

```
[LAMBDA (LST N)                                                  ; Edited 3-Feb-87 16:53 by jop
```

;; returns the Nth element.

```
(COND
   ((EQ N '|...|)
    (CDR (LAST LST)))
   ((EQ N '&&)
    (NTH LST (ADD1 MAXINSPECTCDRLEVEL)))
   (T (CAR (NTH LST N])
```

(SETNTHTOPLEVELELT

```
[LAMBDA (LST N NEWVALUE)                                         ; Edited 3-Feb-87 16:55 by jop
```

;; sets the nth top level eltment of LST to NEWVALUE

;; undoable but it will almost certainly be undone in the wrong place.

```
(COND
   ((EQ N '|...|)
    (/RPLACD (LAST LST)
           NEWVALUE))
   ((EQ N '&&)
    (PROMPTPRINT "Can't set the tail.")                          ; return current value for printing.
    (NTH LST (ADD1 MAXINSPECTCDRLEVEL)))
   (T (PROG NIL
            (RETURN (/RPLACA (OR (NTH LST N)
                                 (RETURN))
                             NEWVALUE])
```

(DEDITE

```
[LAMBDA (EXPR WHERE)                                             ; Edited 24-Sep-87 09:50 by jop
  (LET ((*EDITMODE* 'DISPLAY))
       (EDITE EXPR NIL NIL NIL NIL '(:DONTWAIT :DISPLAY])
```

(FINDRECDECL

```
[LAMBDA (DATUM)                                                  ; Edited 3-Feb-87 16:49 by jop
```

;; find the datatype declaration for a datum.

```
(PROG (TYPENAME DECL)
      (RETURN (AND [SETQ DECL (RECLOOK (SETQ TYPENAME (COND
                                                         ((LISTP DATUM)
```

```
                                                        (CAR DATUM))
                                                       (T (TYPENAME DATUM]
                        (TYPENAMEP DATUM TYPENAME)
                        DECL])
```

(FINDSYSRECDECL

```
  [LAMBDA (DATUM)                                                    ; Edited  3-Feb-87 16:49 by jop
    ;; find the datatype declaration for a if it is a system datatype.
    (PROG (TYPENAME DECL)
          (AND (SETQ TYPENAME (TYPENAME DATUM))
               (SETQ DECL (SYSRECLOOK1 TYPENAME))
               (TYPENAMEP DATUM TYPENAME)
               (RETURN DECL])
```

(MAKE-INSPECTOR-PROFILE

```
  [LAMBDA (NAME)                                                     ; Edited  4-Feb-87 15:35 by jop
    (LET ((P-NAME (OR NAME "INSPECTOR PROFILE")))
         (XCL:MAKE-PROFILE P-NAME '(XCL:*EVAL-FUNCTION* XCL:*EVAL-FUNCTION*)
                '(*PRINT-CASE* *PRINT-CASE*)
                '(*READTABLE* *READTABLE*)
                '(*PACKAGE* *PACKAGE*])
```

(CONFIRM-SET

```
  [LAMBDA NIL                                                        ; Edited  7-Aug-87 09:53 by jop
    (MOUSECONFIRM "This is a potentially dangerous operation."])

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS INSPECTMACROS INSPECTALLFIELDSFLG INSPECTDONTSORTFIELDS SetPropertyMenu SetStackMenu InspectMenu
       PropertyLeftMargin MaxValueLeftMargin INSPECTPRINTLEVEL InspectBitmapMenu ItemWCommandMenu
       InspectPropsMenu MAXINSPECTARRAYLEVEL MAXINSPECTCDRLEVEL MaxInspectorWindowWidth MaxInspectorWindowHeight
       INSPECT.HUNK.COMMANDS USERRECLST SYSPROPS IT MinSpaceBetweenProperyAndValue
       MaxInspectorPropertyValueWidth)
)

(RPAQ? INSPECTDONTSORTFIELDS T)

(RPAQ? INSPECTALLFIELDSFLG T)

(RPAQ? MaxInspectorWindowWidth 330)

(RPAQ? MaxInspectorWindowHeight 606)

(RPAQQ INSPECTPRINTLEVEL (2 . 5))

;; To deal with profiles in spawned processes

(DECLARE%: EVAL@COMPILE

(PUTPROPS EVAL.AS.PROCESS.WITH.PROFILE MACRO [ARGS (LET ((PROFILE (CAR ARGS))
                                                         (FORM (CADR ARGS)))
                                                        `(EVAL.AS.PROCESS (LIST 'XCL:WITH-PROFILE
                                                                                (LIST 'QUOTE ,PROFILE)
                                                                                ,FORM])

(PUTPROPS WITH-INSPECTOR-ENV MACRO [ARGS (LET ((PROFILE (CAR ARGS))
                                               (FORMS (CDR ARGS)))
                                              `(XCL:WITH-PROFILE ,PROFILE (LET ((*PRINT-LEVEL* (CAR
                                                                                                 INSPECTPRINTLEVEL
                                                                                                 ))
                                                                                (*PRINT-LENGTH* (CDR
                                                                                                 INSPECTPRINTLEVEL
                                                                                                 )))
                                                                               ,@FORMS])
)

;; Atom inspector

(DEFINEQ

(INSPECT/ATOM
  [LAMBDA (ATM ALWAYSASKFLG WHERE TAG)                               ; Edited 12-Sep-2022 20:59 by rmk
                                                                     ; Edited  1-Sep-87 10:47 by woz
    ;; asks which aspect to inspect and inspects it.
    (LET ((ASPECTS (TYPESOF ATM NIL NIL '?))
          (OFFER-INSPECT-CODE? (CCODEP ATM))
          (PROFILE (MAKE-INSPECTOR-PROFILE))
          TYPETOINSPECT)
         [COND
```

```
            ((NONSYSPROPNAMES ATM)                                  ; add the property list to selectable aspects.
             (push ASPECTS 'PROPS))
            ((AND (NULL ASPECTS)
                  (GETPROPLIST ATM))                                ; If there is nothing else to inspect about this atom, offer its
                                                                    ; propertylist.
             (SETQ ASPECTS '(PROPS]
          [COND
            ((AND (MEMB 'VARS ASPECTS)
                  (LITATOM (EVALV ATM)))                            ; break the loop that can result from inspecting something that
                                                                    ; has an atom as its value
             (SETQ ASPECTS (REMOVE 'VARS ASPECTS]
          (COND
            ((NOT ASPECTS)
             (PRINTOUT PROMPTWINDOW T ATM " does not have any aspect to inspect.")
             NIL)
            ((EQUAL ASPECTS '(VARS))
             (INSPECT (EVALV ATM)))
            ([SETQ TYPETOINSPECT (COND
                                   ((AND (NULL (CDR ASPECTS))
                                         (EQ (CAR ASPECTS)
                                             'PROPS))
                                    ;; if there is only one aspect and determining how to inspect that aspect gives the user a chance to
                                    ;; quit, don't force a selection at the aspect level.
                                    'PROPS)
                                   (T (SELECT.ATOM.ASPECT ATM ALWAYSASKFLG ASPECTS OFFER-INSPECT-CODE?]
             ;; the functions applyed by this EVAL must evalaute their arguments.  EDITF works because it is happy to take (QUOTE FN) as an
             ;; argument too.
             (SELECTQ TYPETOINSPECT
                  (PROPS                                            ; ask what method to use to inspect it.
                         (SELECTQ [MENU (COND
                                          ((type? MENU InspectPropsMenu)
                                           InspectPropsMenu)
                                          (T (SETQ InspectPropsMenu (create MENU
                                                                            ITEMS _ '(("EDITP" :EDITP "Calls
                                                                                              EDITP on the atom.")
                                                                                      ("Inspect Props" :INSPECT
                                                                                              "Inspects the
                                                                                              property list with
                                                                                              an inspect window."]
                                  (:EDITP ;; IL:EDITP is an NLambda yuk NoSpread yuk
                                          [EVAL.AS.PROCESS.WITH.PROFILE PROFILE `(EDITP ,ATM])
                                  (:INSPECT [EVAL.AS.PROCESS.WITH.PROFILE PROFILE `(INSPECT/PROPLIST
                                                                                    ',ATM NIL ',WHERE])
                                  NIL))
                  (:INSPECTCODE (INSPECTCODE ATM WHERE))
                  (EVAL.AS.PROCESS.WITH.PROFILE PROFILE `(ED ',ATM '(:DONTWAIT :DISPLAY ,TYPETOINSPECT])
```

(SELECT.ATOM.ASPECT

```
[LAMBDA (ATOM ALWAYSASKFLG ASPECTS OFFER-INSPECT-CODE?)             ; Edited 1-Sep-87 10:48 by woz
```

;; Returns a file package type name corresponding to the type of ATOM. The user is asked to choose if there is more than one or If
;; ALWAYSASKFLG is non-NIL. If OFFER-INSPECT-CODE? is set then let Inspect Code be an option in the menu, and return :INSPECTCODE to
;; let the caller know that code rather than filemanager definition is wanted.

```
(LET [(ASPECTS (OR ASPECTS (TYPESOF ATOM NIL NIL '?]
     (COND
        ((NULL ASPECTS)
         NIL)
        ((OR ALWAYSASKFLG (CDR ASPECTS))                            ; ASPECTS is in menu item format
         (MENU (create MENU
                       ITEMS _ (CL:IF OFFER-INSPECT-CODE?
                                   (CONS '("Inspect Code" :INSPECTCODE "Shows the compiled code.")
                                         ASPECTS)
                                   ASPECTS)
                       TITLE _ (CONCAT "Which defn of " ATOM "?")
                       CENTERFLG _ T)))
        (T (CAR ASPECTS])
```

(INSPECT/AS/FUNCTION

```
[LAMBDA (ATM STKP WINDOW)                                           ; Edited 19-Sep-95 13:57 by sybalsky:mv:envos
```

;; calls an editor on function ATM. STKP and WINDOW are the stack pointer and window of the break in which this inspect command was called

```
(LET ((EDITOR (SELECT.FNS.EDITOR ATM))
      FRAME CODEBASE PROC)
     (AND EDITOR (if (EQ EDITOR 'INSPECTCODE)
                     then (COND
                            ([AND (STACKP STKP)
                                  (NOT (fetch (FX INVALIDP) of (SETQ FRAME (fetch (STACKP EDFXP) of STKP]
                             (INSPECTCODE (COND
                                            ((EQ (\GET-COMPILED-CODE-BASE ATM)
                                                 (SETQ CODEBASE (fetch (FX FNHEADER) of FRAME)))
                                             ATM)
                                            (T
```

```
                                        ;; Function executing in this frame is not the one in the definition cell of its name, so fetch
                                        ;; the real code.  Have to pass a CCODEP
                                                    (MAKE-COMPILED-CLOSURE CODEBASE)))
                                            NIL NIL NIL (fetch (FX PC) of FRAME)))
                                     (T (INSPECTCODE ATM)))
                              else (LET [[PROC (AND WINDOW (WINDOWPROP WINDOW 'PROCESS]
                                         (EDITORARGS (if (EQ EDITOR 'ED)
                                                         then (LIST ATM '(METHOD-FNS FUNCTIONS FNS :DONTWAIT :DISPLAY))
                                                       else (LIST ATM]
                                    (if PROC
                                        then (PROCESS.APPLY PROC EDITOR EDITORARGS)
                                      else (CL:APPLY EDITOR EDITORARGS])
```

(SELECT.FNS.EDITOR

```
  [LAMBDA (FN)                                                    ; Edited  1-Sep-87 10:49 by woz
    ;; gives the user a menu choice of editors.  Return the name of the editor function to apply.
    (MENU (create MENU
                  ITEMS _ [APPEND [COND
                                     ((CCODEP FN)
                                      '(("Inspect Code" 'INSPECTCODE "Shows the compiled code."]
                                   '(("Display Edit" 'ED "Edit it with the display editor")
                                     ("Tty Edit" 'EF "Edit it with the standard editor"]
                  CENTERFLG _ T])

)

;; Compiled code inspector

(DEFINEQ

(INSPECTCODE
  [LAMBDA (FN WHERE LVFLG RADIX PC CODEPRINTER)                   ; Edited  4-Feb-87 15:41 by jop
    ;; creates a window that shows the compiled code of a function.
    (COND
       ((GETD 'OPENTEXTSTREAM)                                    ; Use smarter inspector
        (\TEDIT.INSPECTCODE FN WHERE LVFLG RADIX PC CODEPRINTER))
       (T (COND
             ((NOT (CCODEP FN))
              (ERROR "Not a compiled function" FN)))
          (LET [(WINDOW (DECODE.WINDOW.ARG WHERE 400 320 (CONCAT FN " Code Window"]
               (WINDOWPROP WINDOW 'DATUM FN)
               (WINDOWPROP WINDOW 'REPAINTFN (FUNCTION \INSPECT/CODE/REPAINTFN))
               (WINDOWPROP WINDOW 'RESHAPEFN (FUNCTION \INSPECT/CODE/RESHAPEFN))
               (WINDOWPROP WINDOW 'SCROLLFN (FUNCTION SCROLLBYREPAINTFN))
               (WINDOWPROP WINDOW 'PROFILE (MAKE-INSPECTOR-PROFILE))
                                                                  ; call the reshapefn to note the upper left corner and the extent.
               (\INSPECT/CODE/RESHAPEFN WINDOW])

(\TEDIT.INSPECTCODE
  [LAMBDA (FN WHERE LVFLG RADIX PC CODEPRINTER)                   ; Edited 11-Oct-2021 14:04 by rmk:
    (PROG ((STREAM (OPENSTREAM '{NODIRCORE} 'BOTH))
           WINDOW SEL)
          (APPLY* (OR CODEPRINTER (FUNCTION PRINTCODE))
                 FN LVFLG RADIX STREAM NIL PC)
          [SETQ STREAM (OPENTEXTSTREAM STREAM [SETQ WINDOW (DECODE.WINDOW.ARG
                                                            WHERE 400 280 (COND
                                                                             ((OR (LITATOM FN)
                                                                                  (NOT (CCODEP FN)))
                                                                              (CONCAT "Code for " FN))
                                                                             (T (CONCAT (COND
                                                                                           (PC "Code for frame "
                                                                                               )
                                                                                           (T "CCODEP named "))
                                                                                        (fetch (COMPILED-CLOSURE
                                                                                                FRAMENAME)
                                                                                          of FN]
                                              NIL NIL `(READONLY T PROMPTWINDOW DON'T FONT ,DEFAULTFONT]
          (COND
             ((AND PC (SETQ SEL (TEDIT.FIND STREAM "----------" 1)))
                                                                  ; Highlight location of PC
              (TEDIT.SETSEL STREAM (IMAX 1 (IDIFFERENCE SEL 100))
                     0
                     'LEFT)
              (TEDIT.NORMALIZECARET STREAM)))
          [COND
             ((DEFINEDP 'TEXTICON)                                ; Override TEdit's icon
              (WINDOWPROP WINDOW 'ICONFN (FUNCTION TEXTICON]
          (RETURN FN])

(\INSPECT/CODE/RESHAPEFN
  [LAMBDA (WIN OLDIMAGE OLDREGION)                                ; Edited  3-Feb-87 15:35 by jop
```

```
    ;; reshapes a code inspection window.
    ;; set the upper left corner for the repaintfn, call the repaintfn and note the Y position for the extent.
    (PROG [WHEIGHT BOTTOM (FONT (fetch DDFONT of (fetch IMAGEDATA of (WINDOWPROP WIN 'DSP]
          [WINDOWPROP WIN 'REGIONUPPERLEFT (create POSITION
                                                   XCOORD _ 0
                                                   YCOORD _ (SUB1 (IDIFFERENCE (SETQ WHEIGHT (WINDOWPROP
                                                                                              WIN
                                                                                              'HEIGHT))
                                                                               (FONTPROP FONT 'ASCENT]
          (\INSPECT/CODE/REPAINTFN WIN)
          (WINDOWPROP WIN 'EXTENT (create REGION
                                          LEFT _ 0
                                          BOTTOM _ [SETQ BOTTOM (IPLUS (DSPYPOSITION NIL WIN)
                                                                       (FONTPROP FONT 'ASCENT]
                                          WIDTH _ (WINDOWPROP WIN 'WIDTH)
                                          HEIGHT _ (IDIFFERENCE WHEIGHT BOTTOM])
```

(\INSPECT/CODE/REPAINTFN

```
  [LAMBDA (WIN)                                                    ; Edited  8-Apr-87 16:40 by jop
    ;; moves to the window's upper left corner and prints the code for the function in WIN.
    (WITH-INSPECTOR-ENV (WINDOWPROP WIN 'PROFILE)
           (PROG [(UPPERLEFT (WINDOWPROP WIN 'REGIONUPPERLEFT]
                 (MOVETO (fetch (POSITION XCOORD) of UPPERLEFT)
                         (fetch (POSITION YCOORD) of UPPERLEFT)
                         WIN)
            ;; should be changed to pass WIN as a parameter when PRINTCODE is changed to take file argument.
                 (PRINTCODE (WINDOWPROP WIN 'DATUM)
                         NIL 8 WIN])

)

;; Hash table inspector

(DEFINEQ

(INSPECT/HARRAYP
  [LAMBDA (HARRAY WHERE TAG)                                       ; Edited 12-Sep-2022 20:57 by rmk
                                                                   ; Edited  2-Feb-87 17:06 by jop
    ;; opens an inspect window onto the elements of HARRAY
    (PROG ((PROPS (HARRAYKEYS HARRAY)))
          (RETURN (COND
                     (PROPS (INSPECTW.CREATE HARRAY (FUNCTION HARRAYKEYS)
                                             (FUNCTION INSPECTW.GETHASH)
                                             (FUNCTION INSPECTW.PUTHASH)
                                             NIL NIL NIL NIL NIL WHERE NIL TAG))
                     (T (PROMPTPRINT "No keys in that Hash array.")
                        NIL])

(HARRAYKEYS
  [LAMBDA (HARRAY)                                                 ; Edited  3-Feb-87 16:50 by jop
    ;; returns a list of all of the keys in a Hash array.
    (PROG (ITEMLST)
          [MAPHASH HARRAY (FUNCTION (LAMBDA (HASHEDVALUE HASHITEM)
                                      (SETQ ITEMLST (CONS HASHITEM ITEMLST]
          (RETURN ITEMLST])

(INSPECTW.GETHASH
  [LAMBDA (HARRAY ITEM)                                            ; Edited  3-Feb-87 16:51 by jop
    ;; version of GETHASH that switches the order of arguments.
    (GETHASH ITEM HARRAY])

(INSPECTW.PUTHASH
  [LAMBDA (HARRAY ITEM VALUE)                                      ; Edited  3-Feb-87 16:52 by jop
    ;; version of PUTHASH that switches the order of arguments.
    (/PUTHASH ITEM VALUE HARRAY])

)

;; Readtable, termtable inspectors

(DEFINEQ

(RDTBL\NONOTHERCODES
  [LAMBDA (RT)                                                     ; Edited 10-Jul-2021 20:31 by rmk:
                                                                   ; Edited  3-Feb-87 16:54 by jop
```

;; returns the character codes that are not OTHER.

```
(LET (RESULT)
     (DECLARE (SPECVARS RESULT))
     (\MAPCHARTABLE [FUNCTION (LAMBDA (VAL KEY)
                                (CL:WHEN (NEQ (GETSYNTAX KEY RT)
                                              'OTHER)
                                         (PUSH RESULT KEY]
            (fetch READSA of (\GTREADTABLE RT T)))
```

;; (RECORDFIELDNAMES 'READTABLEP)

```
(APPEND '(CASEINSENSITIVE COMMONLISP COMMONNUMSYNTAX DISPATCHMACRODEFS ESCAPECHAR ESCAPEFLG
                HASHMACROCHAR MULTESCAPECHAR NUMBERBASE PACKAGECHAR READMACRODEFS READMACROFLG READSA
                READTBLNAME USESILPACKAGE)
        (SORT RESULT])
```

(GETSYNTAXPROP

```
[LAMBDA (RDTBL CH/FIELD)                                    ; Edited 10-Jul-2021 20:17 by rmk:
```

;; version of GETSYNTAX that has arguments in the right order for inspector, and also shows field names as well as character assignments

```
(SELECTQ CH/FIELD
     (READSA (fetch (READTABLEP READSA) of RDTBL))
     (READMACRODEFS
          (fetch (READTABLEP READMACRODEFS) of RDTBL))
     (READMACROFLG (fetch (READTABLEP READMACROFLG) of RDTBL))
     (ESCAPEFLG (fetch (READTABLEP ESCAPEFLG) of RDTBL))
     (COMMONLISP (fetch (READTABLEP COMMONLISP) of RDTBL))
     (NUMBERBASE (fetch (READTABLEP NUMBERBASE) of RDTBL))
     (CASEINSENSITIVE
          (fetch (READTABLEP CASEINSENSITIVE) of RDTBL))
     (COMMONNUMSYNTAX
          (fetch (READTABLEP COMMONNUMSYNTAX) of RDTBL))
     (USESILPACKAGE
          (fetch (READTABLEP USESILPACKAGE) of RDTBL))
     (DISPATCHMACRODEFS
          (fetch (READTABLEP DISPATCHMACRODEFS) of RDTBL))
     (HASHMACROCHAR
          (fetch (READTABLEP HASHMACROCHAR) of RDTBL))
     (ESCAPECHAR (fetch (READTABLEP ESCAPECHAR) of RDTBL))
     (MULTESCAPECHAR
          (fetch (READTABLEP MULTESCAPECHAR) of RDTBL))
     (PACKAGECHAR (fetch (READTABLEP PACKAGECHAR) of RDTBL))
     (READTBLNAME (fetch (READTABLEP READTBLNAME) of RDTBL))
     (GETSYNTAX CH/FIELD RDTBL])
```

(SETSYNTAXPROP

```
[LAMBDA (RDTBL CH/FIELD VALUE)                              ; Edited 10-Jul-2021 20:20 by rmk:
```

;; version of SETSYNTAX that has arguments in the right order for inspector, plus allows setting fields.

```
(SELECTQ CH/FIELD
     (READSA (replace (READTABLEP READSA) of RDTBL with VALUE))
     (READMACRODEFS
          (replace (READTABLEP READMACRODEFS) of RDTBL with VALUE))
     (READMACROFLG (replace (READTABLEP READMACROFLG) of RDTBL with VALUE))
     (ESCAPEFLG (replace (READTABLEP ESCAPEFLG) of RDTBL with VALUE))
     (COMMONLISP (replace (READTABLEP COMMONLISP) of RDTBL with VALUE))
     (NUMBERBASE (replace (READTABLEP NUMBERBASE) of RDTBL with VALUE))
     (CASEINSENSITIVE
          (replace (READTABLEP CASEINSENSITIVE) of RDTBL with VALUE))
     (COMMONNUMSYNTAX
          (replace (READTABLEP COMMONNUMSYNTAX) of RDTBL with VALUE))
     (USESILPACKAGE
          (replace (READTABLEP USESILPACKAGE) of RDTBL with VALUE))
     (DISPATCHMACRODEFS
          (replace (READTABLEP DISPATCHMACRODEFS) of RDTBL with VALUE))
     (HASHMACROCHAR
          (replace (READTABLEP HASHMACROCHAR) of RDTBL with VALUE))
     (ESCAPECHAR (replace (READTABLEP ESCAPECHAR) of RDTBL with VALUE))
     (MULTESCAPECHAR
          (replace (READTABLEP MULTESCAPECHAR) of RDTBL with VALUE))
     (PACKAGECHAR (replace (READTABLEP PACKAGECHAR) of RDTBL with VALUE))
     (READTBLNAME (replace (READTABLEP READTBLNAME) of RDTBL with VALUE))
     (SETSYNTAX CH/FIELD VALUE RDTBL])
```

(GETTTBLPROP

```
[LAMBDA (TTBL PROP)                                         ; Edited  3-Feb-87 16:50 by jop
```

;; inspector function that returns the value of the property from a terminal table. Combines several miscellaneous parts of the terminal table into a
;; uniform interface.

```
(COND
   ((NUMBERP PROP)
    (ECHOCONTROL PROP NIL TTBL))
   ((FMEMB PROP '(CHARDELETE WORDDELETE LINEDELETE RETYPE CTRLV EOL))
    (CAR (GETSYNTAX PROP TTBL)))
```

```
       ((FMEMB PROP '(1STCHDEL NTHCHDEL POSTCHDEL EMPTYCHDEL))
        (DELETECONTROL PROP NIL TTBL))
       ((EQ PROP 'LINEDELETESTR)
        (DELETECONTROL 'LINEDELETE NIL TTBL))
       ((EQ PROP 'ECHODELS?)
        (EQ (GETDELETECONTROL 'ECHO TTBL)
            'ECHO))
       ((EQ PROP 'CONTROL)
        (GETCONTROL TTBL))
       ((EQ PROP 'RAISE)
        (GETRAISE TTBL))
       ((EQ PROP 'ECHOMODE)
        (GETECHOMODE TTBL])
```

(**SETTTBLPROP**

```
  [LAMBDA (TTBL PROP NEWVALUE)                                    ; Edited  3-Feb-87 16:55 by jop
```

;; inspector function that sets the value of the property from a terminal table. Combines several miscellaneous parts of the terminal table into a ;; uniform interface.

```
    (COND
       ((NUMBERP PROP)
        (ECHOCONTROL PROP NEWVALUE TTBL))
       ((FMEMB PROP '(CHARDELETE WORDDELETE LINEDELETE RETYPE CTRLV EOL))
        (SETSYNTAX NEWVALUE PROP TTBL))
       ((FMEMB PROP '(1STCHDEL NTHCHDEL POSTCHDEL EMPTYCHDEL))
        (DELETECONTROL PROP NEWVALUE TTBL))
       ((EQ PROP 'LINEDELETESTR)
        (DELETECONTROL 'LINEDELETE NEWVALUE TTBL))
       ((EQ PROP 'ECHODELS?)
        (DELETECONTROL (COND
                          (NEWVALUE 'ECHO)
                          (T 'NOECHO))
              NIL TTBL))
       ((EQ PROP 'CONTROL)
        (CONTROL NEWVALUE TTBL))
       ((EQ PROP 'RAISE)
        (RAISE NEWVALUE TTBL))
       ((EQ PROP 'ECHOMODE)
        (ECHOMODE NEWVALUE TTBL])

)

(ADDTOVAR INSPECTMACROS
          (READTABLEP RDTBL\NONOTHERCODES GETSYNTAXPROP SETSYNTAXPROP)
          (TERMTABLEP (CHARDELETE WORDDELETE LINEDELETE RETYPE CTRLV EOL RAISE ECHOMODE LINEDELETESTR 1STCHDEL
                              NTHCHDEL POSTCHDEL EMPTYCHDEL ECHODELS? CONTROL 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
                              15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31)
                GETTTBLPROP SETTTBLPROP))

;; Hunk inspector

(DEFINEQ
```

(**INSPECT/AS/BLOCKRECORD**

```
  [LAMBDA (INSTANCE WHERE CHOICES TAG)                            ; Edited 12-Sep-2022 21:04 by rmk
                                                                  ; Edited  3-Feb-87 16:50 by jop
```

;; offers the user a choice of record types to inspect INSTANCE with.

```
    (LET (RECNAME)
         (COND
            ([NULL (OR CHOICES (SETQ CHOICES (LIST-ALL-BLOCKRECORDS]
             (printout PROMPTWINDOW T "Can't Inspect " INSTANCE))
            ([SETQ RECNAME (MENU (create MENU
                                         ITEMS _ CHOICES
                                         WHENHELDFN _ (FUNCTION (LAMBDA (ITEM)
                                                                  (PROMPTPRINT "Will inspect the list as if it
                                                                         were a " ITEM]
             (INSPECT INSTANCE RECNAME WHERE TAG])
```

(**INSPECT/TYPELESS**

```
  [LAMBDA (ITEM WHERE TAG)                                        ; Edited 12-Sep-2022 21:01 by rmk
                                                                  ; Edited  2-Feb-87 17:08 by jop
```

;; Inspects an object that is typeless. Check very carefully to see if it might be an arrayblock, in which case we can try to inspect it as some kind of ;; array. Otherwise, we might be able to interpret it as some block record.

```
    (LET (HDR TRLR)
         (COND
            ((AND (type? ARRAYBLOCK ITEM)
                  [\VALIDADDRESSP (SETQ HDR (\ADDBASE ITEM (IMINUS \ArrayBlockHeaderWords]
                  (EQ (fetch (ARRAYBLOCK PASSWORD) of HDR)
                      \ArrayBlockPassword)
                  (fetch (ARRAYBLOCK INUSE) of HDR)
                  (\VALIDADDRESSP (SETQ TRLR (fetch (ARRAYBLOCK TRAILER) of HDR)))
                  (EQ (fetch (ARRAYBLOCK PASSWORD) of TRLR)
                      \ArrayBlockPassword))
```

```
            (INSPECT/HUNK ITEM WHERE (fetch (ARRAYBLOCK GCTYPE) of HDR)
                   (IDIFFERENCE (UNFOLD (fetch (ARRAYBLOCK ARLEN) of HDR)
                                        WORDSPERCELL)
                                \ArrayBlockOverheadWords)
                   TAG))
          (T (INSPECT/AS/BLOCKRECORD ITEM WHERE NIL TAG])
```

(LIST-ALL-BLOCKRECORDS

```
[LAMBDA NIL                                                    (* bvm%: "16-Jun-86 11:22")
  (for RECDEC in USERRECLST when (EQ (CAR RECDEC)
                                     'BLOCKRECORD)
     collect (CADR RECDEC])
```

(INSPECT/HUNK

```
[LAMBDA (DATUM WHERE GCTYPE SIZE TAG)                          ; Edited 12-Sep-2022 20:54 by rmk
                                                               ; Edited  7-Aug-87 10:07 by jop
```

;; Inspects a typeless DATUM, which is either a hunk or an array block, with indicated GCTYPE and SIZE in words.

```
(PROG (ELTSPEC BLOCKRECS)
      [SELECTC GCTYPE
           (CODEBLOCK.GCT                                      ; Compiled code lives here
               (RETURN (INSPECTCODE (INSPECT/MAKE/CCODEP DATUM)
                                    WHERE)))
           (PTRBLOCK.GCT                                       ; Pointers live here, so size is unambiguous
                     (SETQ ELTSPEC '(32 \INSPECT.FETCH.PTR \INSPECT.STORE.PTR)))
           (PROGN                                              ; Completely unboxed, so we don't know how to interpret it
                  (COND
                     ([NULL (SETQ ELTSPEC (MENU (create MENU
                                                        ITEMS _ (COND
                                                             ((SETQ BLOCKRECS (LIST-ALL-BLOCKRECORDS))
                                                              (CONS '("As BLOCKRECORD" 'BLOCKRECORD)
                                                                    INSPECT.HUNK.COMMANDS))
                                                             (T INSPECT.HUNK.COMMANDS))
                                                        CENTERFLG _ T]
                      (RETURN NIL))
                     ((EQ ELTSPEC 'BLOCKRECORD)
                      (RETURN (INSPECT/AS/BLOCKRECORD DATUM WHERE BLOCKRECS TAG]
```

;;; At this point ELTSPEC is a list of (itemsize fetchfn storefn). Create an inspector that inspects the appropriate number of items, based on the size

```
          (INSPECTW.CREATE DATUM (for I from 0 to (IMIN (SUB1 (IQUOTIENT (UNFOLD SIZE BITSPERWORD)
                                                                         (CAR ELTSPEC)))
                                                        MAXINSPECTARRAYLEVEL)
                                       collect I)
                (CADR ELTSPEC)
                (CADDR ELTSPEC)
                NIL NIL NIL NIL NIL WHERE NIL TAG])
```

(\INSPECT.DATATYPE.RAW.FETCH

```
[LAMBDA (INSTANCE FIELD DESCRS)                                ; Edited  3-Feb-87 16:55 by jop
```

;; Used to fetch fields of datatype where we have only the field descriptors, not the original user declaration

```
(FETCHFIELD (CAR (NTH DESCRS FIELD))
       INSTANCE])
```

(\INSPECT.FETCH.8

```
[LAMBDA (INSTANCE FIELD)                                       (* bvm%: "16-Jun-86 11:35")
  (\GETBASEBYTE INSTANCE FIELD])
```

(\INSPECT.FETCH.32

```
[LAMBDA (INSTANCE FIELD)                                       (* bvm%: "16-Jun-86 11:35")
  (\GETBASEFIXP INSTANCE (UNFOLD FIELD WORDSPERCELL])
```

(\INSPECT.FETCH.CHAR

```
[LAMBDA (INSTANCE FIELD)                                       (* bvm%: "16-Jun-86 11:36")
  (CHARACTER (\GETBASEBYTE INSTANCE FIELD])
```

(\INSPECT.FETCH.FATCHAR

```
[LAMBDA (INSTANCE FIELD)                                       (* bvm%: "16-Jun-86 11:36")
  (CHARACTER (\GETBASE INSTANCE FIELD])
```

(\INSPECT.FETCH.PTR

```
[LAMBDA (INSTANCE FIELD)                                       (* bvm%: "16-Jun-86 13:53")
  (\GETBASEPTR INSTANCE (UNFOLD FIELD WORDSPERCELL])
```

(\INSPECT.STORE.8

```
[LAMBDA (INSTANCE FIELD NEWVALUE)                              ; Edited  7-Aug-87 10:04 by jop
  (if (CONFIRM-SET)
```

```
        then (UNDOSAVE (LIST '\INSPECT.STORE.8 INSTANCE FIELD (\GETBASEBYTE INSTANCE FIELD)))
             (\PUTBASEBYTE INSTANCE FIELD NEWVALUE])
```

(\INSPECT.STORE.16

```
  [LAMBDA (INSTANCE FIELD NEWVALUE)                          ; Edited 7-Aug-87 10:27 by jop
    (if (CONFIRM-SET)
        then (UNDOSAVE (LIST '\INSPECT.STORE.16 INSTANCE FIELD (\GETBASE INSTANCE FIELD)))
             (\PUTBASE INSTANCE FIELD NEWVALUE])
```

(\INSPECT.STORE.32

```
  [LAMBDA (INSTANCE FIELD NEWVALUE)                          ; Edited 7-Aug-87 10:05 by jop
    (if (CONFIRM-SET)
        then (UNDOSAVE (LIST '\INSPECT.STORE.32 INSTANCE FIELD (\INSPECT.FETCH.32 INSTANCE FIELD)))
             (\PUTBASEFIXP INSTANCE (UNFOLD FIELD WORDSPERCELL)
                    NEWVALUE])
```

(\INSPECT.STORE.CHAR

```
  [LAMBDA (INSTANCE FIELD NEWVALUE)                          ; Edited 7-Aug-87 10:05 by jop
    (if (CONFIRM-SET)
        then (UNDOSAVE (LIST '\INSPECT.STORE.8 INSTANCE FIELD (\GETBASEBYTE INSTANCE FIELD)))
             (\PUTBASEBYTE INSTANCE FIELD (CHARCODE.DECODE NEWVALUE])
```

(\INSPECT.STORE.FATCHAR

```
  [LAMBDA (INSTANCE FIELD NEWVALUE)                          ; Edited 7-Aug-87 10:27 by jop
    (if (CONFIRM-SET)
        then (UNDOSAVE (LIST '\INSPECT.STORE.16 INSTANCE FIELD (\GETBASE INSTANCE FIELD)))
             (\PUTBASE INSTANCE FIELD (CHARCODE.DECODE NEWVALUE])
```

(\INSPECT.STORE.PTR

```
  [LAMBDA (INSTANCE FIELD NEWVALUE)                          ; Edited 7-Aug-87 10:27 by jop
    (if (CONFIRM-SET)
        then (UNDOSAVE (LIST '\INSPECT.STORE.PTR INSTANCE FIELD (\GETBASEPTR INSTANCE FIELD)))
             (\RPLPTR INSTANCE (UNFOLD FIELD WORDSPERCELL)
                    NEWVALUE])
```

(INSPECT/MAKE/CCODEP

```
  [LAMBDA (CODE)                                             (* bvm%: " 7-Jul-86 16:25")
    (MAKE-COMPILED-CLOSURE CODE])

)

(RPAQ? INSPECT.HUNK.COMMANDS
       '[("As 8-bit array" '(8 \GETBASEBYTE \INSPECT.STORE.8))
         ("As 16-bit array" '(16 \GETBASE \INSPECT.STORE.16))
         ("As 32-bit array" '(32 \INSPECT.FETCH.32 \INSPECT.STORE.32))
         ("As Character array" '(8 \INSPECT.FETCH.CHAR \INSPECT.STORE.CHAR))
         ("As Fat Character array" '(16 \INSPECT.FETCH.FATCHAR \INSPECT.STORE.FATCHAR])

(PUTPROPS INSPECT COPYRIGHT ("Venue & Xerox Corporation" 1982 1983 1984 1985 1986 1987 1990 1991 1993 1995 1999
                                    2018 2021))
```

FUNCTION INDEX

VARIABLE INDEX

MACRO INDEX

RECORD INDEX