

File created: 31-Dec-93 12:26:35 {DSK}<king>export>lispcore>sources>CMLARRAYINSPECTOR.;2

changes to: (FILES TWODINSPECTOR)  
(VARS CMLARRAYINSPECTORCOMS)  
(FNS ICMLARRAY ICMLARRAY.GETREGIONFN ICMLARRAY.GETMENUWGROUP)

previous date: 17-Aug-90 14:15:43 {DSK}<king>export>lispcore>sources>CMLARRAYINSPECTOR.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;  
;; Copyright (c) 1985, 1986, 1987, 1990, 1993 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ **CMLARRAYINSPECTORCOMS**

[;; Inspector for Common-Lisp arrays.

;; Functions used to compute load-time constants later (so must come first!):

```
(FNS \CREATE.INSPECTABLEMENU \CREATE.SETABLEMENU \CREATE.TITLEMENU)
(FNS CREATEARRAYSLICE GET.MENU.LIST ICMLARRAY ICMLARRAY.ATTACHDISPLAY ICMLARRAY.DETACHDISPLAY
  ICMLARRAY.DOWINDOWCOMFN ICMLARRAY.INDICES ICMLARRAY.SETVALUE ICMLARRAY.TITLECOMMANDFN
  ICMLARRAY.VALUECOMMANDFN ICMLARRAY.DISPLAYSLICE ICMLARRAY.GETREGIONFN ICMLARRAY.GETMENUWGROUP
  ICMLARRAY.MENUW.APPLY ICMLARRAY.MENUW.GETLEVEL ICMLARRAY.MENUW.SHOW SLICEDIMENSION SLICERANK
  SLICEREF SLICESSET ZEROD.FETCHFN ZEROD.STOREFN)
[ADDVARS (INSPECTMACROS ((FUNCTION CL:ARRAYP) . ICMLARRAY)
  (INITRECORDS ICML.ARRAYSLICE)
  (FILES TWODINSPECTOR FREEMENU)
  (DECLARE%: EVAL@COMPILE DONTCOPY (RECORDS ICML.ARRAYSLICE))
  (DECLARE%: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY (LOCALVARS . T))
  (DECLARE%: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
    (NLAML)
    (LAMA SLICESSET SLICEREF
      ICMLARRAY.VALUECOMMANDFN)]))
```

;; Inspector for Common-Lisp arrays.

;; Functions used to compute load-time constants later (so must come first!):

(DEFINEQ

(\CREATE.INSPECTABLEMENU

```
[LAMBDA NIL
  (create MENU
    ITEMS _ '(("Inspect" 'INSPECT "Inspect element")
      ("Set" 'SET "Set element")
      ("Indices" 'INDICES "Display indices")
      ("IT _ Selection" 'SETIT "Bind IT to element"])
```

(\CREATE.SETABLEMENU

```
[LAMBDA NIL
  (create MENU
    ITEMS _ '(("Set" 'SET "Set element")
      ("Indices" 'INDICES "Display indices")
      ("IT _ Selection" 'SETIT "Bind IT to element"])
```

(\CREATE.TITLEMENU

```
[LAMBDA NIL
  (create MENU
    ITEMS _ '(("Refetch" 'REFETCH "Refetch the array")
      ("IT _ Datum" 'IT "Bind IT to the inspected array"])
```

)

(DEFINEQ

(CREATEARRAYSLICE

[LAMBDA (CMLARRAY LEVELS) (\* jop%: "22-May-86 11:53")

(\* \* An ARRAYSLICE is a zero, one or two dimensional slice of a CMLARRAY.  
LEVELS is a list of length (CL:ARRAY-RANK CMLARRAY) which describes the slice.  
The atom ALL indicates that that dimension is unrestricted)

```
(LET* ((RANK (CL:ARRAY-RANK CMLARRAY))
  (DIMS (CL:ARRAY-DIMENSIONS CMLARRAY))
  (OFFSETCONSTANT 0)
  (SCANDIMS (bind (PROD _ 1)
    RESULT for DIM in (REVERSE DIMS) do (push RESULT PROD)
    (SETQ PROD (ITIMES PROD DIM))
    finally (RETURN RESULT)))
  SELECTIONDIMS OFFSETS)
[for LEVEL in LEVELS as DIM in DIMS as SCANDIM in SCANDIMS
```

```

do (if (EQ LEVEL 'ALL)
      then (push SELECTIONDIMS DIM)
          (push OFFSETS SCANDIM)
      else (SETQ OFFSETCONSTANT (IPLUS OFFSETCONSTANT (ITIMES LEVEL SCANDIM)
(create ICML.ARRAYSLICE
      SELECTEDDIMS _ (DREVERSE SELECTIONDIMS)
      OFFSETS _ (DREVERSE OFFSETS)
      OFFSETCONSTANT _ OFFSETCONSTANT
      LINEARIZEDARRAY _ (%FLATTEN-ARRAY CMLARRAY))

```

**(GET.MENU.LIST**

```

[LAMBDA (CMLARRAY DISPLAYEDLEVELS MAXWIDTH FONT BFONT) ; Edited 5-Apr-87 18:05 by jop
(LET*
  [(RANK (CL:ARRAY-RANK CMLARRAY))
   (MENU-P (AND (IGREATERP RANK 1)
                 (for DIM in (CL:ARRAY-DIMENSIONS CMLARRAY) always (NEQ DIM 0]
   `((PROPS FONT ,FONT)
    , [if MENU-P
      then `((TYPE MOMENTARY LABEL "SHOW" FONT ,BFONT BOX 1 SELECTEDFN ICMLARRAY.MENUW.SHOW)
              (TYPE MOMENTARY LABEL "APPLY" FONT ,BFONT BOX 1 SELECTEDFN ICMLARRAY.MENUW.APPLY]
      ((GROUP (PROPS FORMAT TABLE)
        ((TYPE DISPLAY LABEL "Element type:")
         (TYPE DISPLAY LABEL , (MKSTRING (CL:ARRAY-ELEMENT-TYPE CMLARRAY))
          FONT
          ,BFONT))
        , @ (IF (SIMPLE-ARRAY-P CMLARRAY)
          THEN `([(TYPE DISPLAY LABEL "Simple-p:")
                   (TYPE DISPLAY LABEL T FONT ,BFONT]
          ELSE `([ (IF (CL:ADJUSTABLE-ARRAY-P CMLARRAY)
                     THEN `([(TYPE DISPLAY LABEL "Adjustable-p:")
                              (TYPE DISPLAY LABEL T FONT ,BFONT]
                     ELSEIF (EXTENDABLE-ARRAY-P CMLARRAY)
                     THEN `([(TYPE DISPLAY LABEL "Extendable-p:")
                              (TYPE DISPLAY LABEL T FONT ,BFONT]
                     , @ (IF (CL:ARRAY-HAS-FILL-POINTER-P CMLARRAY)
                     THEN `([(TYPE DISPLAY LABEL "Fill-pointer-p:")
                              (TYPE DISPLAY LABEL T FONT ,BFONT]
                     , @ (IF (DISPLACED-ARRAY-P CMLARRAY)
                     THEN `([(TYPE DISPLAY LABEL "Displaced-p:")
                              (TYPE DISPLAY LABEL T FONT ,BFONT]
        ((TYPE DISPLAY LABEL "Rank:")
         (TYPE DISPLAY LABEL ,RANK FONT ,BFONT))
        , @ (if (ILESSP RANK 2)
          then `([(TYPE DISPLAY LABEL "Total-size:")
                  (TYPE DISPLAY LABEL , (CL:ARRAY-TOTAL-SIZE CMLARRAY)
                   FONT
                   ,BFONT]
          else `([(TYPE DISPLAY LABEL "Dimension:")
                  , @ (for I from 0 to (SUB1 RANK) collect ` (TYPE DISPLAY LABEL , I FONT ,BFONT]
                  ((TYPE DISPLAY LABEL "Levels:")
                   , @ (for I from 0 to (SUB1 RANK) collect ` (TYPE DISPLAY LABEL , (CL:ARRAY-DIMENSION CMLARRAY
                                                                I)
                                                                FONT
                                                                ,BFONT]
        , @ (if MENU-P
          then `([(TYPE DISPLAY LABEL "Shown:")
                  , @ (for LEVEL in DISPLAYEDLEVELS as I from 0
                    collect ` (TYPE MOMENTARY ID , (PACK* 'LEVEL I)
                              LABEL
                              , LEVEL FONT ,BFONT MAXWIDTH , MAXWIDTH BOX 1 DIM , I SELECTEDFN
                              ICMLARRAY.MENUW.GETLEVEL])

```

**(ICMLARRAY**

```

[LAMBDA (CMLARRAY ASTYPE WHERE) ; Edited 5-Apr-87 17:26 by jop

```

```

;; Top level entry point into the CMLARRAY inspector

```

```

(LET* ((RANK (CL:ARRAY-RANK CMLARRAY))
       (FONT (DEFAULTFONT 'DISPLAY))
       (DISPLAYEDLEVELS (bind (LESS1RANK _ (SUB1 RANK)) for I from 0 to (SUB1 RANK)
                              collect (if (ILESSP (IDIFFERENCE LESS1RANK I)
                                              2)
                                           then 'ALL
                                           else 0)))
       (DISPLAYGROUP MENUGROUP TOPLEFT)
  [if (for DIM in (CL:ARRAY-DIMENSIONS CMLARRAY) always (IGREATERP DIM 0))
      then (SETQ DISPLAYGROUP (ICMLARRAY.DISPLAYSLICE CMLARRAY DISPLAYEDLEVELS WHERE))
          (SETQ TOPLEFT (create POSITION
                                XCOORD _ (ADD1 (fetch (REGION RIGHT) of (WINDOWREGION DISPLAYGROUP)))
                                YCOORD _ (fetch (REGION TOP) of (WINDOWREGION DISPLAYGROUP]
          (SETQ MENUGROUP (ICMLARRAY.GETMENUWGROUP CMLARRAY FONT DISPLAYEDLEVELS TOPLEFT))
          (if DISPLAYGROUP
              then (ICMLARRAY.ATTACHDISPLAY DISPLAYGROUP MENUGROUP DISPLAYEDLEVELS))
          MENUGROUP])

```

**(ICMLARRAY.ATTACHDISPLAY**

```
[LAMBDA (DISPLAYGROUP STATUSGROUP DISPLAYEDLEVELS) (* jop%: "24-Nov-85 15:45")
  (ATTACHWINDOW DISPLAYGROUP STATUSGROUP 'LEFT 'TOP)
  (for w in (CONS DISPLAYGROUP (ALLATTACHEDWINDOWS DISPLAYGROUP)) do (WINDOWPROP W 'DOWINDOWCOMFN
    (FUNCTION ICMLARRAY.DOWINDOWCOMFN))
  )
  (WINDOWPROP STATUSGROUP 'DISPLAYGROUP DISPLAYGROUP)
  (WINDOWPROP STATUSGROUP 'CURRENTLEVELS DISPLAYEDLEVELS])
```

**(ICMLARRAY.DETACHDISPLAY**

```
[LAMBDA (STATUSGROUP) (* jop%: " 4-Oct-85 17:53")

  (**)

  (PROG [(DISPLAYGROUP (WINDOWPROP STATUSGROUP 'DISPLAYGROUP)
    (DETACHWINDOW DISPLAYGROUP)
    (CLOSEW DISPLAYGROUP])
```

**(ICMLARRAY.DOWINDOWCOMFN**

```
[LAMBDA (WINDOW) (* jop%: "24-Nov-85 15:45")

  (** Pass on the usual comms, except for SHAPEW)

  (PROG ((PASSTOMAINCOMS (WINDOWPROP WINDOW 'PASSTOMAINCOMS))
    (COM (MENU WindowMenu)))
    (if COM
      then (LET* [(CENTRAL (CENTRALWINDOW WINDOW))
        (DISPLAYGROUP (WINDOWPROP CENTRAL 'DISPLAYGROUP)
        (if (EQ COM 'SHAPEW)
          then [SHAPEW DISPLAYGROUP (GETREGION NIL NIL NIL (FUNCTION ICMLARRAY.GETREGIONFN)
            (CONS DISPLAYGROUP 'CLOSED)]
          elseif (MEMB COM PASSTOMAINCOMS)
            then (APPLY* COM CENTRAL)
            else (APPLY* COM WINDOW])
```

**(ICMLARRAY.INDICES**

```
[LAMBDA (DISPLAYWINDOW ROW COLUMN) ; Edited 5-Apr-87 17:11 by jop

  ;; Display the indices of the selected item

  (LET* [(MAINW (MAINWINDOW DISPLAYWINDOW))
    (CURRENTLEVELS (WINDOWPROP MAINW 'CURRENTLEVELS))
    (PRTWINDOW (WINDOWPROP MAINW 'PRTWINDOW)
    (PRINTOUT PRTWINDOW T "Indices: ")

    (bind FIRSTFLG for LEVEL in CURRENTLEVELS
      do (if (EQ LEVEL 'ALL)
        then (if FIRSTFLG
          then (PRINTOUT PRTWINDOW %, COLUMN %,)
          else (SETQ FIRSTFLG T)
            (PRINTOUT PRTWINDOW %, ROW %,))
        else (PRINTOUT PRTWINDOW %, LEVEL %,)])

    ; In the zero-d case ROW and COLUMN are NIL. In the one-d
    ; case COLUMN is NIL
```

**(ICMLARRAY.SETVALUE**

```
[LAMBDA (DISPLAYWINDOW ROW COLUMN) ; Edited 8-Apr-87 16:47 by jop

  ;; In the zero and one-d cases COLUMN should be NIL, and ROW is the only index

  (PROG ((MAINW (MAINWINDOW DISPLAYWINDOW))
    [SLICERANK (SLICERANK (WINDOWPROP DISPLAYWINDOW 'DATUM)
    PRTWINDOW NEWVALUE)
    (SETQ PRTWINDOW (WINDOWPROP MAINW 'PRTWINDOW))
    (WITH-INSPECTOR-ENV (WINDOWPROP DISPLAYWINDOW 'PROFILE)
      (RESETLST
        (RESETSAVE (TTYDISPLAYSTREAM PRTWINDOW))
        (RESETSAVE (TTY.PROCESS (THIS.PROCESS)))
        (CLEARBUF T T)
        (PRINTOUT T T "Eval> ")
        (SETQ NEWVALUE (CL:FUNCALL XCL:*EVAL-FUNCTION* (LISPXREAD T T)))
        ; clear tty buffer because it sometimes has stuff left.
        (CLEARBUF T T)))
    (if (EQL SLICERANK 2)
      then (TWODINSPECT.REPLACE DISPLAYWINDOW ROW COLUMN NEWVALUE)
      else (ONEDINSPECT.REPLACE DISPLAYWINDOW ROW NEWVALUE])
```

**(ICMLARRAY.TITLECOMMANDFN**

```
[LAMBDA (WINDOW) ; Edited 20-Jul-90 20:02 by yabu
  (if (MOUSESTATE MIDDLE)
    then (LET* ((TITLEMENU (CONSTANT (\CREATE.TITLEMENU)))
      ; Original was (create MENU ITEMS _ '("Refetch" 'REFETCH
      ; "Refetch the array") ("IT _ Datum" 'IT "Bind IT to the inspected
      ; array"))).
      ; Changed by yabu.fx, for SUNLOADUP without DWIM.

      (CMLARRAY (WINDOWPROP (MAINWINDOW WINDOW)
        'CMLARRAY))
```

```

(MENUW (MAINWINDOW WINDOW)))
(SELECTQ (MENU TITLEMENU)
  (REFETCH (ICMLARRAY.MENUW.SHOW (FM.GETITEM 'SHOW NIL MENUW)
    MENUW)
    (LET [(DISPLAYGROUP (WINDOWPROP MENUW 'DISPLAYGROUP))
      (TOPRIGHT (with REGION (WINDOWPROP MENUW 'REGION)
        (create POSITION
          XCOORD _ (SUB1 LEFT)
          YCOORD _ TOP)))
      (LEVELS (WINDOWPROP MENUW 'CURRENTLEVELS)
        (ICMLARRAY.DETACHDISPLAY MENUW)
        (SETQ DISPLAYGROUP (XCL:WITH-PROFILE (WINDOWPROP DISPLAYGROUP 'PROFILE)
          (ICMLARRAY.DISPLAYSLICE CMLARRAY LEVELS
            DISPLAYGROUP TOPRIGHT)))
        (ICMLARRAY.ATTACHDISPLAY DISPLAYGROUP MENUW LEVELS)))
      (IT (SETQ IT CMLARRAY)
        (PROMPTPRINT "IT bound to " CMLARRAY))
      NIL])

```

**(ICMLARRAY.VALUECOMMANDFN**

```

[LAMBDA ARGS
  (PROG ((INSPECTABLEMENU (CONSTANT (\CREATE.INSPECTABLEMENU)))
    ; Edited 20-Jul-90 19:59 by yabu
    ; Original was (create MENU ITEMS _ '("Inspect" 'INSPECT
    ; "Inspect element") ("Set" 'SET "Set element") ("Indices"
    ; 'INDICES "Display indices") ("IT _ Selection" 'SETIT "Bind IT to
    ; element"))).
    ; Changed by yabu.fx, for SUNLOADUP without DWIM.
    ; Original was (create MENU ITEMS _ '("Set" 'SET "Set
    ; element") ("Indices" 'INDICES "Display indices") ("IT _
    ; Selection" 'SETIT "Bind IT to element"))).
    ; Changed by yabu.fx, for SUNLOADUP without DWIM.

    (SETABLEMENU (CONSTANT (\CREATE.SETABLEMENU)))

    (VALUE (ARG ARGS 1))
    INDEX ROW COLUMN SLICE DISPLAYWINDOW SLICERANK)
  (if (EQL ARGS 4)
    then
      (SETQ INDEX (ARG ARGS 2))
      (SETQ SLICE (ARG ARGS 3))
      (SETQ DISPLAYWINDOW (ARG ARGS 4))
      ; must be in the one-d case
    else
      (SETQ ROW (ARG ARGS 2))
      (SETQ COLUMN (ARG ARGS 3))
      (SETQ SLICE (ARG ARGS 4))
      (SETQ DISPLAYWINDOW (ARG ARGS 5))
      ; must be in the two-d case
      (SETQ SLICERANK (SLICERANK SLICE))
      (SELECTQ (if (OR (NUMBERP VALUE)
        (NULL VALUE))
        then (MENU SETABLEMENU)
        else (MENU INSPECTABLEMENU))
        (INSPECT (INSPECT VALUE))
        (SET (SELECTQ SLICERANK
          (0 (ICMLARRAY.SETVALUE DISPLAYWINDOW INDEX))
          (1 (ICMLARRAY.SETVALUE DISPLAYWINDOW INDEX))
          (2 (ICMLARRAY.SETVALUE DISPLAYWINDOW ROW COLUMN))
          (SHOULDNT)))
          (SETIT (SETQ IT (SELECTQ SLICERANK
            (0 (SLICEREF SLICE))
            (1 (SLICEREF SLICE INDEX))
            (2 (SLICEREF SLICE ROW COLUMN))
            (SHOULDNT)))
            ; Nice to have some feedback
            (PROMPTPRINT (CONCAT "IT bound to " VALUE)))
          (INDICES (SELECTQ SLICERANK
            (0 (ICMLARRAY.INDICES DISPLAYWINDOW))
            (1 (ICMLARRAY.INDICES DISPLAYWINDOW INDEX))
            (2 (ICMLARRAY.INDICES DISPLAYWINDOW ROW COLUMN))
            (SHOULDNT)))
            NIL])

```

**(ICMLARRAY.DISPLAYSLICE**

```

[LAMBDA (CMLARRAY LEVELS WHERE TOPRIGHT)
  (LET ((SLICE (CREATEARRAYSLICE CMLARRAY LEVELS)))
    (SELECTQ (SLICERANK SLICE)
      (0 (ONEDINSPECTW.CREATE SLICE '("Entry")
        (FUNCTION ZEROD.FETCHFN)
        (FUNCTION ZEROD.STOREFN)
        (FUNCTION ICMLARRAY.VALUECOMMANDFN)
        NIL "Display Window" (FUNCTION ICMLARRAY.TITLECOMMANDFN)
        WHERE TOPRIGHT))
      (1 (ONEDINSPECTW.CREATE SLICE (for I from 0 to (SUB1 (SLICEDIMENSION SLICE 0)) collect I)
        (FUNCTION SLICEREF)
        (FUNCTION SLICESSET)
        (FUNCTION ICMLARRAY.VALUECOMMANDFN)
        NIL "Display Window" (FUNCTION ICMLARRAY.TITLECOMMANDFN)
        WHERE TOPRIGHT))
      (2 (TWOEDINSPECTW.CREATE SLICE (for I from 0 to (SUB1 (SLICEDIMENSION SLICE 0)) collect I)
        (for I from 0 to (SUB1 (SLICEDIMENSION SLICE 1)) collect I)

```

; Edited 5-Apr-87 17:15 by jop

```

(FUNCTION SLICEREF)
(FUNCTION SLICESET)
(FUNCTION ICMLARRAY.VALUECOMMANDFN)
NIL NIL "Display Window" (FUNCTION ICMLARRAY.TITLECOMMANDFN)
WHERE TOPRIGHT))
(SHOULDNT "Should not happen"))

```

**(ICMLARRAY.GETREGIONFN**

[LAMBDA (FIXEDPOINT MOVINGPOINT INFO)

; Edited 5-Apr-87 17:26 by jop

;; Controlled reshape of a CMLARRAY inspector display window. For use with GETREGION Assumes that info is CONS pair (WINDOW . STATE)  
 ;; The initial state is CLOSED. Assumes no init region or minsize

```

(PROG ((WINDOW (CAR INFO))
      (STATE (CDR INFO))
      WINDOWREGION)
  (SETQ WINDOWREGION (WINDOWREGION WINDOW))
  (if (NULL MOVINGPOINT)
      then [RETURN (create POSITION
                           XCOORD _ (ADD1 (fetch (REGION RIGHT) of WINDOWREGION))
                           YCOORD _ (ADD1 (fetch (REGION TOP) of WINDOWREGION))]
      else (if (EQ STATE 'CLOSED)
               then (RPLACD INFO 'OPEN)
                   [RETURN (create POSITION
                                XCOORD _ (SUB1 (fetch (REGION LEFT) of WINDOWREGION))
                                YCOORD _ (SUB1 (fetch (REGION BOTTOM) of WINDOWREGION))]
               else (if (IGREATERP (fetch (POSITION XCOORD) of MOVINGPOINT)
                                (fetch (REGION RIGHT) of WINDOWREGION))
                       then (replace (POSITION XCOORD) of MOVINGPOINT with (fetch (REGION RIGHT) of WINDOWREGION))
                       )
               (if (IGREATERP (fetch (POSITION YCOORD) of MOVINGPOINT)
                                (fetch (REGION TOP) of WINDOWREGION))
                   then (replace (POSITION YCOORD) of MOVINGPOINT with (fetch (REGION TOP) of WINDOWREGION))
                   (RETURN MOVINGPOINT)]))

```

**(ICMLARRAY.GETMENUWGROUP**

[LAMBDA (CMLARRAY FONT DISPLAYEDLEVELS TOPLEFT)

; Edited 5-Apr-87 17:25 by jop

;; Constructs the three windows of the status group and puts them up on the screen. returns the mainwindow of the group.

```

(LET* ((BFONT (FONTCREATE (FONTPROP FONT 'FAMILY)
                          (FONTPROP FONT 'SIZE)
                          'BRR))
      (DIMS (CL:ARRAY-DIMENSIONS CMLARRAY))
      (RANK (CL:ARRAY-RANK CMLARRAY))
      [PHEIGHT (HEIGHTIFWINDOW (FONTPROP FONT 'HEIGHT)
                                SWINDOW PWINDOW)
      (SETQ SWINDOW (FREEMENU (GET.MENU.LIST CMLARRAY DISPLAYEDLEVELS
                                           (IMAX (STRINGWIDTH 'ALL BFONT)
                                           (STRINGWIDTH (for DIM in DIMS largest (STRINGWIDTH DIM BFONT))
                                           BFONT))
                                           FONT BFONT)
                             (RESETVAR *PRINT-ARRAY* NIL (CONCAT CMLARRAY " Inspector"))
                             ; Makes no sense to reshape the statuswindow group
                             ; Cache the datum
                             ; DISPLAYEDLEVELS is a description of the array slice to be
                             ; displayed
                             ; PWINDOW is the prompt window
      (if (for DIM in DIMS always (NEQ DIM 0))
          then (SETQ PWINDOW (CREATEW (CREATEREGION 0 0 100 PHEIGHT)
                                       NIL NIL T))
              (WINDOWPROP PWINDOW 'MINSIZE (CONS 0 PHEIGHT))
              (WINDOWPROP PWINDOW 'MAXSIZE (CONS MAX.SMALLP PHEIGHT))
              (WINDOWPROP PWINDOW 'PAGEFULLFN (FUNCTION NIL))
              (DSPSCROLL 'ON PWINDOW)
              (WINDOWPROP SWINDOW 'PRTWINDOW PWINDOW)
              (DSPFONT FONT PWINDOW))
          ; position and open the windowgroup
      [MOVEW SWINDOW (if TOPLEFT
                        then [create POSITION
                                XCOORD _ (fetch (POSITION XCOORD) of TOPLEFT)
                                YCOORD _ (IDIFFERENCE (fetch (POSITION YCOORD) of TOPLEFT)
                                                       (SUB1 (fetch (REGION HEIGHT) of (WINDOWPROP SWINDOW 'REGION))
                                                       (fetch (REGION HEIGHT) of (WINDOWPROP SWINDOW 'REGION))
                                else (GETBOXPOSITION (fetch (REGION WIDTH) of (WINDOWPROP SWINDOW 'REGION))
                                                       (fetch (REGION HEIGHT) of (WINDOWPROP SWINDOW 'REGION))
      (REDISPLAYW SWINDOW)
      (if PWINDOW
          then (ATTACHWINDOW PWINDOW SWINDOW 'BOTTOM))
      SWINDOW])

```

**(ICMLARRAY.MENUW.APPLY**

[LAMBDA (ITEM MENUWINDOW BUTTONS)

; Edited 5-Apr-87 17:28 by jop

;; Display the slice described by the windowprop LEVELS

```

(LET* [(CMLARRAY (WINDOWPROP MENUWINDOW 'CMLARRAY))
      (DISPLAYGROUP (WINDOWPROP MENUWINDOW 'DISPLAYGROUP))

```

```

(TOPRIGHT (with REGION (WINDOWPROP MENUWINDOW 'REGION)
              (create POSITION
                XCOORD _ (SUB1 LEFT)
                YCOORD _ TOP)))
(LEVELS (for I from 0 to (SUB1 (CL:ARRAY-RANK CMLARRAY)) collect (FM.ITEMPROP (FM.GETITEM
                                                                 (PACK* 'LEVEL I)
                                                                 NIL MENUWINDOW)
                                                                 'LABEL]

(if (IGREATERP (for LEVEL in LEVELS count (EQ LEVEL 'ALL))
    2)
    then (PRINTOUT (WINDOWPROP MENUWINDOW 'PRTWINDOW)
      T "Rank too high")
    else (ICMLARRAY.DETACHDISPLAY MENUWINDOW)
      (SETQ DISPLAYGROUP (XCL:WITH-PROFILE (WINDOWPROP DISPLAYGROUP 'PROFILE)
        (ICMLARRAY.DISPLAYSLICE CMLARRAY LEVELS DISPLAYGROUP TOPRIGHT)))
      (ICMLARRAY.ATTACHDISPLAY DISPLAYGROUP MENUWINDOW LEVELS])

```

**(ICMLARRAY.MENUW.GETLEVEL**

[LAMBDA (ITEM MENUWINDOW BUTTONS)

; Edited 5-Apr-87 17:28 by jop

;; Get a new LEVEL for dim DIM

```

(LET ((DIM (FM.ITEMPROP ITEM 'DIM))
      (LEVEL (FM.ITEMPROP ITEM 'LABEL))
      (CMLARRAY (WINDOWPROP MENUWINDOW 'CMLARRAY))
      (PRTWINDOW (WINDOWPROP MENUWINDOW 'PRTWINDOW))
      (LEVMENU NEWVALUE))
  (SETQ LEVEL (if (ILESSP (CL:ARRAY-DIMENSION CMLARRAY DIM)
    10)
    then (LET [(LEVMENU (OR (FM.ITEMPROP ITEM 'LEVMENU)
      (create MENU
        ITEMS _ (CONS ' (ALL 'ALL "Unrestricted")
          (for I from 0
            to (SUB1 (CL:ARRAY-DIMENSION CMLARRAY
              DIM))
            collect (LIST I (KWOTE I)

          (FM.ITEMPROP ITEM 'LEVMENU LEVMENU)
          (OR (MENU LEVMENU)
            LEVEL))
        else (PRINTOUT PRTWINDOW T)
          (RESETFORM (TTY.PROCESS (THIS.PROCESS))
            (SETQ NEWVALUE (PROMPTFORWARD "New level?" LEVEL (CONCAT "Type new level for
              dim " DIM)
              PRTWINDOW)))
          (if (STRINGP NEWVALUE)
            then (if (STREQUAL (U-CASE NEWVALUE)
              "ALL")
              then 'ALL
              else (SETQ NEWVALUE (READ (OPENSTRINGSTREAM NEWVALUE)))
                (if (AND (FIXP NEWVALUE)
                  (GEQ NEWVALUE 0)
                  (LESSP NEWVALUE (CL:ARRAY-DIMENSION CMLARRAY DIM)))
                  then NEWVALUE
                  else (PRINTOUT (WINDOWPROP MENUWINDOW 'PRTWINDOW)
                    T
                    (CONCAT "Illegal value " NEWVALUE))
                    LEVEL))
              else LEVEL)))
      (FM.CHANGELABEL ITEM LEVEL MENUWINDOW])

```

**(ICMLARRAY.MENUW.SHOW**

[LAMBDA (ITEM MENUWINDOW BUTTONS)

; Edited 7-Apr-87 10:25 by jop

```

(LET [(DISPLAYEDLEVELS (WINDOWPROP MENUWINDOW 'CURRENTLEVELS))
      (CMLARRAY (WINDOWPROP MENUWINDOW 'CMLARRAY))
      (bind LEVEL-ITEM for I from 0 to (SUB1 (CL:ARRAY-RANK CMLARRAY)) as LEVEL in DISPLAYEDLEVELS
        do (SETQ LEVEL-ITEM (FM.GETITEM (PACK* 'LEVEL I)
          NIL MENUWINDOW))
      (if LEVEL-ITEM
        then (FM.CHANGELABEL LEVEL-ITEM LEVEL MENUWINDOW])

```

**(SLICEDIMENSION**

[LAMBDA (SELECTION DIM)

(\* jop%: "20-Nov-85 20:23")

(\* \*)

```

(CAR (FNTH (fetch (ICML.ARRAYSlice SELECTEDDIMS) of SELECTION)
  (ADD1 DIM])

```

**(SLICERANK**

[LAMBDA (SELECTION)

(\* jop%: "20-Nov-85 20:23")

(\* \*)

```

(LENGTH (fetch (ICML.ARRAYSlice SELECTEDDIMS) of SELECTION])

```

**(SLICEREF**

```

[LAMBDA ARGS
  (if (ILESSP ARGS 1)
    then (HELP "Need at least one arg"))
  (LET* ((SLICE (ARG ARGS 1))
    (LINEARIZEDARRAY (fetch (ICML.ARRAYSLICE LINEARIZEDARRAY) of SLICE))
    (OFFSETS (fetch (ICML.ARRAYSLICE OFFSETS) of SLICE))
    (OFFSETCONSTANT (fetch (ICML.ARRAYSLICE OFFSETCONSTANT) of SLICE)))
    (CL:AREF LINEARIZEDARRAY (IPLUS OFFSETCONSTANT (for OFFSET in OFFSETS as I from 2
      sum (ITIMES OFFSET (ARG ARGS I))

```

; Edited 5-Apr-87 17:11 by jop

**(SLICESET**

```

[LAMBDA ARGS
  (* *)

  (if (ILESSP ARGS 2)
    then (HELP "Need at least two args"))
  (LET* ((NEWVALUE (ARG ARGS 1))
    (SLICE (ARG ARGS 2))
    (LINEARIZEDARRAY (fetch (ICML.ARRAYSLICE LINEARIZEDARRAY) of SLICE))
    (OFFSETS (fetch (ICML.ARRAYSLICE OFFSETS) of SLICE))
    (OFFSETCONSTANT (fetch (ICML.ARRAYSLICE OFFSETCONSTANT) of SLICE)))
    (ASET NEWVALUE LINEARIZEDARRAY (IPLUS OFFSETCONSTANT (for OFFSET in OFFSETS as I from 3
      sum (ITIMES OFFSET (ARG ARGS I))

```

(\* jop%: " 5-Aug-86 12:20")

**(ZEROD.FETCHFN**

```

[LAMBDA (SLICE PROP)
  (* *)

  (SLICEREF SLICE])

```

(\* jop%: " 5-Aug-86 12:20")

**(ZEROD.STOREFN**

```

[LAMBDA (NEWVALUE SLICE PROP)
  (* *)

  (SLICESET NEWVALUE SLICE])

```

(\* jop%: " 5-Aug-86 12:20")

```

)

(ADDTOVAR INSPECTMACROS ((FUNCTION CL:ARRAYP) . ICMLARRAY))

(/DECLAREDATATYPE 'ICML.ARRAYSLICE ' (POINTER POINTER POINTER POINTER)
  ;; ---field descriptor list elided by lister---
  ' 8)

(FILESLoad TWODINSPECTOR FREEMENU)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(DATATYPE ICML.ARRAYSLICE (SELECTEDDIMS OFFSETS OFFSETCONSTANT LINEARIZEDARRAY))
)

(/DECLAREDATATYPE 'ICML.ARRAYSLICE ' (POINTER POINTER POINTER POINTER)
  ;; ---field descriptor list elided by lister---
  ' 8)
)

(DECLARE%: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)
)

(DECLARE%: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVARS

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA SLICESET SLICEREF ICMLARRAY.VALUECOMMANDFN)
)

(PUTPROPS CMLARRAYINSPECTOR COPYRIGHT ("Venue & Xerox Corporation" 1985 1986 1987 1990 1993))

```





FUNCTION INDEX

CREATEARRAYSLICE .....	1	ICMLARRAY.GETMENUWGROUP ..	5	ICMLARRAY.TITLECOMMANDFN	3	ZEROD.STOREFN .....	7
GET.MENU.LIST .....	2	ICMLARRAY.GETREGIONFN ..	5	ICMLARRAY.VALUECOMMANDFN	4	\CREATE.INSPECTABLEMENU ..	1
ICMLARRAY .....	2	ICMLARRAY.INDICES .....	3	SLICEDIMENSION .....	6	\CREATE.SETABLEMENU .....	1
ICMLARRAY.ATTACHDISPLAY ..	3	ICMLARRAY.MENUW.APPLY ..	5	SLICERANK .....	6	\CREATE.TITLEMENU .....	1
ICMLARRAY.DETACHDISPLAY ..	3	ICMLARRAY.MENUW.GETLEVEL	6	SLICEREF .....	7		
ICMLARRAY.DISPLAYSLICE ..	4	ICMLARRAY.MENUW.SHOW ....	6	SLICESET .....	7		
ICMLARRAY.DOWINDOWCOMFN ..	3	ICMLARRAY.SETVALUE .....	3	ZEROD.FETCHFN .....	7		

RECORD INDEX

ICML.ARRAYSLICE .....	7
-----------------------	---

VARIABLE INDEX

INSPECTMACROS .....	7
---------------------	---