

File created: 4-Apr-88 12:02:12 {ERINYES}<LISPUSERS>MEDLEY>PIECE-MENUS.;1

changes to: (FNS CHUNK.MENU.CREATE CHUNK.MENU.GET.REAL.MENU CHUNK.MENU.INVOKE KEYWORD.MENU.CREATE
KEYWORD.MENU.GET.MENU KEYWORD.MENU.INVOKE KEYWORD.MENU.MAKE.MENU PIECE.MENU.MAKE.MENU)

previous date: 12-Feb-86 18:31:27 {PHYLUM}<LISP>KOTO>LISPUSERS>PIECE-MENUS.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1986, 1988 by Xerox Corporation. All rights reserved.

```
(RPAQQ PIECE-MENUSCOMS ((FNS CHUNK.MENU.CREATE CHUNK.MENU.GET.REAL.MENU CHUNK.MENU.INVOKE KEYWORD.MENU.CREATE  
KEYWORD.MENU.GET.MENU KEYWORD.MENU.INVOKE KEYWORD.MENU.MAKE.MENU  
PIECE.MENU.MAKE.MENU)  
(BITMAPS CHUNK.MENU.DOWN.BITMAP CHUNK.MENU.UP.BITMAP KEYWORD.MENU.KEYWORD.BITMAP)  
(DECLARE%: DOEVAL@LOAD (LOCALVARS .T)  
(GLOBALVARS CHUNK.MENU.DOWN.BITMAP CHUNK.MENU.UP.BITMAP  
KEYWORD.MENU.KEYWORD.BITMAP))))
```

(DEFINEQ

(CHUNK.MENU.CREATE

```
[LAMBDA (ITEMS PROPERTIES REQUIRED.ITEMS) ; Edited 4-Apr-88 11:56 by Briggs  
(PROG (BLANK.ITEM UP.ITEM DOWN.ITEM CHUNK.COUNT IT.LISTS ITS N MENU.COUNT BLOCK.ITS ITM STR MENUS)  
(SETQ BLANK.ITEM (LIST " " (KWOTE '$BLANK$)  
"No action"))  
(SETQ UP.ITEM (LIST CHUNK.MENU.UP.BITMAP (KWOTE '$UP$)  
"Jump to preceding section"))  
(SETQ DOWN.ITEM (LIST CHUNK.MENU.DOWN.BITMAP (KWOTE '$DOWN$)  
"Jump to following section"))  
(SETQ CHUNK.COUNT (OR (LISTGET PROPERTIES 'CHUNK.COUNT)  
30))  
(SETQ IT.LISTS (CONS))  
(SETQ ITS (CONS))  
(SETQ N 0)  
(for ITEM in ITEMS do (if (EQ N CHUNK.COUNT)  
then (TCONC IT.LISTS (CAR ITS))  
(SETQ ITS (CONS))  
(SETQ N 0))  
(TCONC ITS ITEM)  
(SETQ N (ADD1 N))  
finally (TCONC IT.LISTS (CAR ITS)))  
(SETQ IT.LISTS (CAR IT.LISTS))  
(SETQ MENU.COUNT (LENGTH IT.LISTS))  
(SETQ BLOCK.ITS (for LST in IT.LISTS as I from 1 collect (SETQ ITM (CAR LST))  
(SETQ STR (if (LISTP ITM)  
then (CAR ITM)  
else ITM))  
(LIST (CONCAT STR "...")  
(LIST 'QUOTE (CONS '$CHUNK$ I))  
(CONCAT "Jump to menu chunk starting with  
item " STR))  
(SETQ MENUS (for LST in IT.LISTS as I from 1  
collect (SETQ ITS (CONS))  
(if REQUIRED.ITEMS  
then (for RIT in REQUIRED.ITEMS do (TCONC ITS RIT))  
(TCONC ITS BLANK.ITEM))  
(if (IGREATERP MENU.COUNT 1)  
then (for BLOCK.ITM in BLOCK.ITS as J from 1  
do (if (EQ J I)  
then (if (NEQ I 1)  
then (TCONC ITS UP.ITEM))  
(if (NEQ I MENU.COUNT)  
then (TCONC ITS DOWN.ITEM))  
else (TCONC ITS BLOCK.ITM))  
(TCONC ITS BLANK.ITEM))  
(SETQ ITS (NCONC (CAR ITS)  
LST))  
(PIECE.MENU.MAKE.MENU ITS PROPERTIES)))  
(RETURN (CONS MENUS 1]))
```

(CHUNK.MENU.GET.REAL.MENU

```
[LAMBDA (CHUNK.MENU) ; Edited 4-Apr-88 11:57 by Briggs  
(PROG (MENUS N)  
(SETQ MENUS (CAR CHUNK.MENU))  
(SETQ N (CDR CHUNK.MENU))  
(RETURN (CAR (NTH MENUS N]))
```

(CHUNK.MENU.INVOKE

```

[LAMBDA (CHUNK.MENU POSITION) ; Edited 4-Apr-88 11:57 by Briggs
  (PROG (MENUS N CURRENT.MENU DONE POS NEW.POSITION RESULT THUMB.ITEMS THUMB.MENU PROPERTIES)
    (SETQ MENUS (CAR CHUNK.MENU))
    (GETMOUSESTATE)
    (SETQ POS (OR POSITION (create POSITION
      XCOORD _ LASTMOUSEX
      YCOORD _ LASTMOUSEY)))
    [until DONE do (SETQ N (CDR CHUNK.MENU))
      (SETQ CURRENT.MENU (CAR (NTH MENUS N)))
      [SETQ NEW.POSITION (create POSITION
        XCOORD _ (IDIFFERENCE (fetch (POSITION XCOORD) of POS)
          (IQUOTIENT (fetch (MENU IMAGEWIDTH) of
            CURRENT.MENU
              )
                2))
        YCOORD _ (IDIFFERENCE (fetch (POSITION YCOORD) of POS)
          (IQUOTIENT (fetch (MENU IMAGEHEIGHT)
            of CURRENT.MENU)
                2))
        ]
      (SETQ RESULT (MENU CURRENT.MENU NEW.POSITION))
      (if (LISTP RESULT)
        then (SELECTQ (CAR RESULT)
          ($CHUNK$ (RPLACD CHUNK.MENU (CDR RESULT)))
          (SETQ DONE T))
        else (SELECTQ RESULT
          ($BLANK$)
          ($UP$ (RPLACD CHUNK.MENU (SUB1 N)))
          ($DOWN$ (RPLACD CHUNK.MENU (ADD1 N)))
          (SETQ DONE T)
        ]
      (RETURN RESULT])

```

(KEYWORD.MENU.CREATE

```

[LAMBDA (OBJECTS KEYWORDFN PROPERTIES ITEMFN) ; Edited 4-Apr-88 11:58 by Briggs
  (PROG (TITLE ALST ENTRY ITEM ITEMS KEYWORD.ITEMS KEYWORD)
    [for OBJECT in OBJECTS do (SETQ ITEM (if ITEMFN
      then (APPLY* ITEMFN OBJECT)
      else OBJECT))
      (for KEYWD in (APPLY* KEYWORDFN OBJECT)
        do (SETQ ENTRY (FASSOC KEYWD ALST))
          (if ENTRY
            then (SETQ ITEMS (CADR ENTRY))
              (NCONC1 ITEMS ITEM)
            else (SETQ ALST (CONS (CONS KEYWD (CONS (LIST ITEM)
              NIL))
                ALST))
          (SETQ ALST (SORT ALST T))
        ]
    [SETQ KEYWORD.ITEMS (for ENT in ALST collect (SETQ KEYWORD (CAR ENT))
      (LIST (CONCAT KEYWORD "'s")
        (KWOTE (CONS '$KEYWORD$ KEYWORD))
        (CONCAT "Jump to section for " KEYWORD))
      (RETURN (LIST (CAAR ALST)
        ALST PROPERTIES KEYWORD.ITEMS])

```

(KEYWORD.MENU.GET.MENU

```

[LAMBDA (ENTRY KEYWORD.MENU) ; Edited 4-Apr-88 11:58 by Briggs
  (OR (CDDR ENTRY)
    (PROG (ITEMS KEYWORD PROPERTIES KEYWORD.ITEMS TITLE)
      (SETQ ITEMS (CADR ENTRY))
      (SETQ KEYWORD (CAR ENTRY))
      (SETQ PROPERTIES (CADDR KEYWORD.MENU))
      (SETQ KEYWORD.ITEMS (CADDR KEYWORD.MENU))
      (RPLACD (CDR ENTRY)
        (CHUNK.MENU.CREATE ITEMS (NCONC (LIST 'TITLE (if (SETQ TITLE (LISTGET PROPERTIES
          'TITLE))
            then (CONCAT TITLE ": " KEYWORD)
            else KEYWORD))
          PROPERTIES)
        KEYWORD.ITEMS))
      (RETURN (CDDR ENTRY])

```

(KEYWORD.MENU.INVOKE

```

[LAMBDA (KEYWORD.MENU POSITION) ; Edited 4-Apr-88 11:59 by Briggs
  (PROG (ALST DONE ENTRY RESULT SUBMENU REALMENU NEW.POS POS)
    (SETQ ALST (CADR KEYWORD.MENU))
    (SETQ POS (if POSITION
      else (GETMOUSESTATE)
      (create POSITION
        XCOORD _ LASTMOUSEX
        YCOORD _ LASTMOUSEY)))
    (until DONE do (SETQ ENTRY (FASSOC (CAR KEYWORD.MENU)
      ALST))
      (SETQ SUBMENU (KEYWORD.MENU.GET.MENU ENTRY KEYWORD.MENU))
      (SETQ REALMENU (CHUNK.MENU.GET.REAL.MENU SUBMENU))
      [SETQ NEW.POS (create POSITION

```

```

XCOORD _ (IDIFFERENCE (fetch (POSITION XCOORD) of POS)
                (IQUOTIENT (fetch (MENU IMAGEWIDTH) of REALMENU)
                2))
YCOORD _ (IDIFFERENCE (fetch (POSITION YCOORD) of POS)
                (IQUOTIENT (fetch (MENU IMAGEHEIGHT) of REALMENU)
                2))
(SETQ RESULT (CHUNK.MENU.INVOKE SUBMENU NEW.POS))
(if (AND (LISTP RESULT)
        (EQ (CAR RESULT)
            '$KEYWORDS$))
    then (RPLACA KEYWORD.MENU (CDR RESULT))
    else (SETQ DONE T))
(RETURN RESULT])

```

(KEYWORD.MENU.MAKE.MENU

```
[LAMBDA (ITEMS TITLE PROPERTIES)
```

; Edited 4-Apr-88 12:00 by Briggs
; Edited 10-AUG-83 17:28 by DAHJr

```
(CHUNK.MENU.CREATE ITEMS (NCONC (LIST 'TITLE TITLE)
                                PROPERTIES])
```

(PIECE.MENU.MAKE.MENU

```
[LAMBDA (ITEMS PROPERTIES)
  (PROG (MENU VALUE)
```

; Edited 4-Apr-88 12:00 by Briggs

```

(SETQ MENU (create MENU
                  ITEMS _ ITEMS))
(AND (SETQ VALUE (LISTGET PROPERTIES 'TITLE))
      (replace (MENU TITLE) of MENU with VALUE))
(AND (SETQ VALUE (LISTGET PROPERTIES 'CENTERFLG))
      (replace (MENU CENTERFLG) of MENU with VALUE))
(AND (SETQ VALUE (LISTGET PROPERTIES 'MENUFONT))
      (replace (MENU MENUFONT) of MENU with VALUE))
(AND (SETQ VALUE (LISTGET PROPERTIES 'ITEMWIDTH))
      (replace (MENU ITEMWIDTH) of MENU with VALUE))
(AND (SETQ VALUE (LISTGET PROPERTIES 'ITEMHEIGHT))
      (replace (MENU ITEMHEIGHT) of MENU with VALUE))
(AND (SETQ VALUE (LISTGET PROPERTIES 'MENUBORDERSIZE))
      (replace (MENU MENUBORDERSIZE) of MENU with VALUE))
(AND (SETQ VALUE (LISTGET PROPERTIES 'MENUOUTLINESIZE))
      (replace (MENU MENUOUTLINESIZE) of MENU with VALUE))
(RETURN MENU])

```

)

```
(RPAQQ CHUNK.MENU.DOWN.BITMAP  )
```

```
(RPAQQ CHUNK.MENU.UP.BITMAP  )
```

```
(RPAQQ KEYWORD.MENU.KEYWORD.BITMAP )
```

```
(DECLARE%: DOEVAL@LOAD
```

```
(DECLARE%: DOEVAL@COMPILE DONTCOPY
```

```
(LOCALVARS .T)
```

)

```
(DECLARE%: DOEVAL@COMPILE DONTCOPY
```

```
(GLOBALVARS CHUNK.MENU.DOWN.BITMAP CHUNK.MENU.UP.BITMAP KEYWORD.MENU.KEYWORD.BITMAP)
```

)

)

```
(PUTPROPS PIECE-MENUS COPYRIGHT ("Xerox Corporation" 1986 1988))
```

FUNCTION INDEX

CHUNK.MENU.CREATE1 CHUNK.MENU.INVOKE1 KEYWORD.MENU.GET.MENU ...2 KEYWORD.MENU.MAKE.MENU ..3
CHUNK.MENU.GET.REAL.MENU 1 KEYWORD.MENU.CREATE2 KEYWORD.MENU.INVOKE2 PIECE.MENU.MAKE.MENU3

VARIABLE INDEX

CHUNK.MENU.DOWN.BITMAP3 CHUNK.MENU.UP.BITMAP3 KEYWORD.MENU.KEYWORD.BITMAP3
