

File created: 27-Jan-87 15:56:46 {ERIS}<IRIS>NEXT>COLOR.;2

changes to: (VARS COLORCOMS EditColorMapHeight EditColorMapWidth DICOLOR.hueMapping DICOLOR.lightnessMapping
DICOLOR.saturationMapping NEWCOLORITEM DICOLOR.hueConstants DICOLOR.achromatic DICOLOR.blue
DICOLOR.bluePurple DICOLOR.bluishGreen DICOLOR.bluishPurple DICOLOR.brown
DICOLOR.brownYellow DICOLOR.brownishRed DICOLOR.brownishYellow DICOLOR.green
DICOLOR.greenBlue DICOLOR.greenishBlue DICOLOR.greenishYellow DICOLOR.orange
DICOLOR.orangeYellow DICOLOR.orangishRed DICOLOR.orangishYellow DICOLOR.purple
DICOLOR.purpleRed DICOLOR.purplishBlue DICOLOR.purplishRed DICOLOR.red DICOLOR.redBrown
DICOLOR.redOrange DICOLOR.reddishBrown DICOLOR.reddishOrange DICOLOR.reddishPurple
DICOLOR.yellow DICOLOR.yellowGreen DICOLOR.yellowishBrown DICOLOR.yellowishGreen
DICOLOR.yellowishOrange DICOLOR.saturationConstants DICOLOR.noSaturation DICOLOR.grayish
DICOLOR.moderate DICOLOR.strong DICOLOR.vivid DICOLOR.lightnessConstants DICOLOR.black
DICOLOR.veryDark DICOLOR.dark DICOLOR.medium DICOLOR.light DICOLOR.veryLight DICOLOR.white)
(FNS DISPLAYCOLORLEVELS DISPLAYHLSLEVELS HLSLEVEL HLSTORGB HLSVALUEFN HLSVALUEFROMLEVEL
LEVELFROMHLSVALUE RAINBOWMAP RGBTOHLS OVERPAINT BITMAPFROMSTRING SHADEBITMAP EDITCOLORMAP
GETCOLOR#FROMUSER GETCOLOR#FROMSCREEN DISPLAYCOLORLEVEL FILLINREGION AREAIFILL CENTEREDLEFT
OUTLINEAREA OUTLINEREGION ADJUSTCOLORMAP SHOWCOLORBLOCKS MAPOFACOLOR CNSMENUINIT CNSTOCSL
CNSTORGB CSLTOCNS DICOLOR.FROM.USER GETCNS HLSTOCSL CSLTOHLS RGBTOCNS DICOLOR.hueN
DICOLOR.hueNvalue DICOLOR.hueNname DICOLOR.lightnessN DICOLOR.lightnessNvalue
DICOLOR.lightnessNname DICOLOR.saturationN DICOLOR.saturationNvalue DICOLOR.saturationNname)
(RECORDS hueRecord lightnessRecord saturationRecord)

previous date: 16-Jan-87 18:20:53 {ERIS}<IRIS>NEXT>COLOR.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;
;; Copyright (c) 1982, 1983, 1985, 1986, 1987 by Xerox Corporation. All rights reserved.

(RPAQQ COLORCOMS

```
(FNS DISPLAYCOLORLEVELS DISPLAYHLSLEVELS HLSLEVEL HLSTORGB HLSVALUEFN HLSVALUEFROMLEVEL
  LEVELFROMHLSVALUE RAINBOWMAP RGBTOHLS)
(FNS OVERPAINT BITMAPFROMSTRING SHADEBITMAP)
(INITVARS (EDITCOLORMAP.WINDOW NIL))
(FNS EDITCOLORMAP EDITCOLORMAP.BUTTONEVENTFN EDITCOLORMAP.REDISPLAYFN EDITCOLORMAP.VALUELEVEL
  EDITCOLORMAP.WINDOWLEVEL CHANGECOLORLEVELS GETCOLOR#FROMUSER GETCOLOR#FROMSCREEN DISPLAYCOLORLEVEL
  FILLINREGION AREAIFILL CENTEREDLEFT OUTLINEAREA OUTLINEREGION)
(FNS ADJUSTCOLORMAP SHOWCOLORBLOCKS MAPOFACOLOR COLORHEXPATTERN)
(VARS EditColorMapHeight EditColorMapWidth (COLOR#MENUSAVE)
  (CONTROLMENUSAVE)
  (EDIT8BITCOLORMAPMENU)
  (EDIT8BITCOLORMAPNUMBERREADER))
(GLOBALVARS COLOR#MENUSAVE CONTROLMENUSAVE EDIT8BITCOLORMAPMENU EDIT8BITCOLORMAPNUMBERREADER
  EditColorMapHeight EditColorMapWidth)
(COMS
```

;;; support for global naming and querying of colors.

```
(FNS CNSMENUINIT CNSTOCSL CNSTORGB CSLTOCNS DICOLOR.FROM.USER GETCNS HLSTOCSL CSLTOHLS RGBTOCNS)
(VARS DICOLOR.hueMapping DICOLOR.lightnessMapping DICOLOR.saturationMapping NEWCOLORITEM)
(INITVARS (COLORNAMEMENU))
(FNS DICOLOR.hueN DICOLOR.hueNvalue DICOLOR.hueNname DICOLOR.lightnessN DICOLOR.lightnessNvalue
  DICOLOR.lightnessNname DICOLOR.saturationN DICOLOR.saturationNvalue DICOLOR.saturationNname)
(DECLARE%: EVAL@LOAD DONTCOPY (*)
  (RECORDS hueRecord lightnessRecord saturationRecord)
  (CONSTANTS * DICOLOR.hueConstants)
  (CONSTANTS * DICOLOR.saturationConstants)
  (CONSTANTS * DICOLOR.lightnessConstants))
(P (CNSMENUINIT))
(FILES LLCOLOR READNUMBER)
(P (SETQ EDITBMENU NIL)
  (MOVD 'ARRAYP 'COLORMAPP])
```

(DEFINEQ

(DISPLAYCOLORLEVELS

```
[LAMBDA (WINDOW RGB) (* kbr%: " 3-Jun-86 19:45")
  (PROG (HLS)
    (DISPLAYCOLORLEVEL WINDOW 'RED (fetch (RGB RED) of RGB)
      (fetch (RGB RED) of RGB))
    (DISPLAYCOLORLEVEL WINDOW 'GREEN (fetch (RGB GREEN) of RGB)
      (fetch (RGB GREEN) of RGB))
    (DISPLAYCOLORLEVEL WINDOW 'BLUE (fetch (RGB BLUE) of RGB)
      (fetch (RGB BLUE) of RGB))
    (SETQ HLS (RGBTOHLS RGB))
    (DISPLAYCOLORLEVEL WINDOW 'HUE (fetch (HLS HUE) of HLS)
      (EDITCOLORMAP.WINDOWLEVEL 'HUE (fetch (HLS HUE) of HLS)))
    (DISPLAYCOLORLEVEL WINDOW 'LIGHTNESS (fetch (HLS LIGHTNESS) of HLS)
      (EDITCOLORMAP.WINDOWLEVEL 'LIGHTNESS (fetch (HLS LIGHTNESS) of HLS)))
    (DISPLAYCOLORLEVEL WINDOW 'SATURATION (fetch (HLS SATURATION) of HLS)
      (EDITCOLORMAP.WINDOWLEVEL 'SATURATION (fetch (HLS SATURATION) of HLS]))
```

(DISPLAYHLSLEVELS

```
[LAMBDA (HLS WIN)

  (DISPLAYHLSLEVEL HLS 'HUE NIL WIN)
  (DISPLAYHLSLEVEL HLS 'LIGHTNESS NIL WIN)
  (DISPLAYHLSLEVEL HLS 'SATURATION NIL WIN)]
```

(* rrb "25-OCT-82 14:08")
 (* displays a hue lightness saturation triple in the edit window.)

(HLSLEVEL

```
[LAMBDA (HLS FIELD NEWLEVEL)

  (SELECTQ FIELD
    (HUE (PROG1 (fetch (HLS HUE) of HLS)
      (AND NEWLEVEL (replace (HLS HUE) of HLS with NEWLEVEL))))
    (LIGHTNESS (PROG1 (fetch (HLS LIGHTNESS) of HLS)
      (AND NEWLEVEL (replace (HLS LIGHTNESS) of HLS with NEWLEVEL))))
    (SATURATION (PROG1 (fetch (HLS SATURATION) of HLS)
      (AND NEWLEVEL (replace (HLS SATURATION) of HLS with NEWLEVEL))))
    (SHOULDNT])
```

(* rrb "25-OCT-82 13:29")
 (* returns the value of the named field from a hue lightness saturation record.)

(HLSTORGB

```
[LAMBDA (HLS LIGHTNESS SATURATION)

  (PROG (HUE M1 M2 RGB)
    (COND
      ((LISTP HLS)
        (SETQ HUE (fetch (HLS HUE) of HLS))
        (SETQ LIGHTNESS (fetch (HLS LIGHTNESS) of HLS))
        (SETQ SATURATION (fetch (HLS SATURATION) of HLS)))
      (T (SETQ HUE HLS)))
    [SETQ M1 (COND
      ((FGREATERP 0.5 LIGHTNESS)
        (FTIMES LIGHTNESS (FPLUS 1.0 SATURATION)))
      (T (FDIFFERENCE (FPLUS LIGHTNESS SATURATION)
        (FTIMES LIGHTNESS SATURATION)
        (SETQ M2 (FDIFFERENCE (FTIMES 2.0 LIGHTNESS)
          M1))
        [SETQ RGB (create RGB
          RED _ (HLSVALUEFN M1 M2 HUE)
          GREEN _ (HLSVALUEFN M1 M2 (IDIFFERENCE HUE 120))
          BLUE _ (HLSVALUEFN M1 M2 (IDIFFERENCE HUE 240))
          (RETURN RGB])
```

(* kbr%: " 3-Jun-86 21:16")
 (* Converts from a hue saturation lightness triple into red green blue triple.
 HUE is in range 0 to 360, lightness and saturation are in the range 0 to 1.0 *)
 (* This algorithm was taken from siggraph vol 13 number 3 August 1979%: Status report on graphics standards planning committee. *)

(HLSVALUEFN

```
[LAMBDA (M1 M2 HUE)

  (SETQ HUE (IMOD HUE 360))
  (FIX (FTIMES (COND
    ((ILESSP HUE 60)
      M1)
    [(ILESSP HUE 120)
      (FPLUS M1 (FTIMES (FQUOTIENT (FDIFFERENCE HUE 60)
        60)
        (FDIFFERENCE M2 M1])
      M2)
    [(ILESSP HUE 240)
      (FPLUS M2 (FTIMES (FQUOTIENT (FDIFFERENCE HUE 240)
        60)
        (FDIFFERENCE M1 M2])
      M1)
    (T M1))
    255])
```

(* kbr%: " 3-Jun-86 20:45")
 (* Internal value function for converting from HLS to RGB.
 *)

(HLSVALUEFROMLEVEL

```
[LAMBDA (HLS LEVEL)

  (SELECTQ HLS
    (HUE (IQUOTIENT (ITIMES LEVEL 360)
      255))
    (FQUOTIENT LEVEL 255])
```

(* rrb "25-OCT-82 13:26")

(* returns the scaled value of the hls marker on a scale from 0 to 255)

(LEVELFROMHLSVALUE

[LAMBDA (HLS LEVEL)

(* rrb "25-OCT-82 14:06")

(* returns the level on a scale from 0 to 255 that this value would have.)

```
(SELECTQ HLS
  (HUE (IQUOTIENT (ITIMES LEVEL 255)
                 360))
  (FIX (FTIMES LEVEL 255)))
```

(RAINBOWMAP

[LAMBDA (NBITS)

(* rrb "21-OCT-82 18:14")

```
[OR NBITS (NULL (COLORDISPLAYP))
  (SETQ NBITS (COLORMAPBITS (SCREENCOLORMAP)
                             (COLORMAPCREATE (COND
                                               [(EQ NBITS 8)
                                                (PROG (MAXINTENSITY MINVISIBLERED MINVISIBLEBLUE MINVISIBLEGREEN NSTEPS REDSTEPSSIZE
                                                       GREENSTEPSSIZE BLUESTEPSSIZE)
              (SETQ MAXINTENSITY 255)
              (SETQ MINVISIBLERED 69)
              (SETQ MINVISIBLEBLUE 38)
              (SETQ MINVISIBLEGREEN 38)
              (SETQ NSTEPS (IQUOTIENT (EXPT 2 NBITS)
                                      8))
```

(* determine how many steps are available for each transition from one color to the next. There are 8 such transitions. red up, green up, red down, blue up, green down, red up, green up, all down)

(* minimum visible intensity values were emperically determined but will differ depending upon the brightness setting of the individual display. They are also diddled to make the numer of steps come out right.)

```
(RETURN (NCONC (for I from MINVISIBLERED to MAXINTENSITY
                by (SETQ REDSTEPSSIZE (IQUOTIENT (IPLUS (IDIFFERENCE MAXINTENSITY
                                                         MINVISIBLERED)
                                                         NSTEPS -2)
                                                  NSTEPS))
              collect (* red up)
                    (LIST I 0 0))
  (for I from MINVISIBLEGREEN to MAXINTENSITY
    by (SETQ GREENSTEPSSIZE (IQUOTIENT (IPLUS (IDIFFERENCE
                                               MAXINTENSITY
                                               MINVISIBLEGREEN)
                                               -1 NSTEPS)
                                       NSTEPS))
      collect (* GREEN UP)
            (LIST 255 I 0))
  (for I from REDSTEPSSIZE to (IDIFFERENCE MAXINTENSITY MINVISIBLERED)
    by REDSTEPSSIZE collect (* red down)
      (LIST (IDIFFERENCE MAXINTENSITY I)
            255 0))
  (CONS '(0 255 0))
  (for I from MINVISIBLEBLUE to MAXINTENSITY
    by (SETQ BLUESTEPSSIZE (IQUOTIENT (IPLUS (IDIFFERENCE
                                               MAXINTENSITY
                                               MINVISIBLEBLUE)
                                               -1 NSTEPS)
                                       NSTEPS))
      collect (* BLUE UP)
            (LIST 0 255 I))
  (for I from GREENSTEPSSIZE to (IDIFFERENCE MAXINTENSITY
                                       MINVISIBLEGREEN)
    by GREENSTEPSSIZE collect (* GREEN down)
      (LIST 0 (IDIFFERENCE MAXINTENSITY I)
            255))
  (CONS '(0 0 255))
  (for I from MINVISIBLERED to MAXINTENSITY by REDSTEPSSIZE
    collect (* red up)
          (LIST I 0 255))
  (for I from MINVISIBLEGREEN to MAXINTENSITY by GREENSTEPSSIZE
    collect (* GREEN UP)
          (LIST 255 I 255))
  (for I from GREENSTEPSSIZE to (IDIFFERENCE MAXINTENSITY
                                       MINVISIBLEGREEN)
    by GREENSTEPSSIZE collect (* all down)
      (LIST (IDIFFERENCE MAXINTENSITY I)
            (IDIFFERENCE MAXINTENSITY I)
            (IDIFFERENCE MAXINTENSITY I)))
  (CONS '(0 0 0)
        (T RAINBOWINTENSITIES))
  NBITS])
```

(RGBTOHLS

[LAMBDA (RGB GREEN BLUE) (* kbr%: " 3-Jun-86 20:13")

(* Converts from a red green blue triple of color information into a hue lightness saturation triple.
*)

(* This algorithm was taken from Procedural Elements for Computer Graphics 1985 page 405 by David F. Rogers *)

```
(PROG (RED CR CG CB M1 M2 LIGHTNESS HLS)
(COND
  ((LISTP RGB)
   (SETQ RED (fetch (RGB RED) of RGB))
   (SETQ GREEN (fetch (RGB GREEN) of RGB))
   (SETQ BLUE (fetch (RGB BLUE) of RGB)))
  (T (SETQ RED RGB)))
(SETQ M1 (MAX RED GREEN BLUE))
(SETQ M2 (MIN RED GREEN BLUE))
(SETQ LIGHTNESS (FQUOTIENT (FPLUS (FQUOTIENT M1 255)
                                  (FQUOTIENT M2 255))
                          2))
[SETQ HLS (COND
  ((EQ M1 M2)
   (create HLS
            HUE _ 0
            LIGHTNESS _ LIGHTNESS
            SATURATION _ 0.0))
  (T (SETQ CR (FQUOTIENT (IDIFFERENCE M1 RED)
                        (IDIFFERENCE M1 M2)))
     (SETQ CG (FQUOTIENT (IDIFFERENCE M1 GREEN)
                        (IDIFFERENCE M1 M2)))
     (SETQ CB (FQUOTIENT (IDIFFERENCE M1 BLUE)
                        (IDIFFERENCE M1 M2)))
     (create HLS
              HUE _ (IMOD (FIX (FTIMES [COND
                                     ((EQ M1 RED)
                                      (FDIFFERENCE CB CG))
                                     ((EQ M1 GREEN)
                                      (FPLUS 2.0 (FDIFFERENCE CR CB)))
                                     (T (FPLUS 4.0 (FDIFFERENCE CG CR)
                                         60.0))
                                     360)
                          LIGHTNESS _ LIGHTNESS
                          SATURATION _ (COND
                                         ((FGREATERP 0.5 LIGHTNESS)
                                          (FQUOTIENT (IDIFFERENCE M1 M2)
                                                      (IPLUS M1 M2)))
                                         (T (FQUOTIENT (IDIFFERENCE M1 M2)
                                                      (IDIFFERENCE (ITIMES 2 255)
                                                                (IPLUS M1 M2)
                                                                ))))
                                         ))))
     (RETURN HLS])
)
```

(DEFINEQ

(OVERPAINT

[LAMBDA (BM1 BM2 X Y TXT SCR) (* kbr%: " 2-Sep-85 20:30")

(* Uses BM1 as a mask thru which it paints the INVERSE of texture onto BM2 at position X Y)

```
(PROG (BMW BMH)
(SETQ BMW (BITMAPWIDTH BM1))
(SETQ BMH (BITMAPHEIGHT BM1))
(OR SCR (SETQ SCR (BITMAPCOPY BM1))) (* We need a scratch BM. Most demos cache one)
(BITBLT BM1 0 0 SCR 0 0 BMW BMH 'INPUT 'REPLACE)
(BITBLT NIL NIL NIL SCR 0 0 BMW BMH 'TEXTURE 'ERASE TXT)
(BITBLT BM1 0 0 BM2 X Y BMW BMH 'INPUT 'ERASE)
(BITBLT SCR 0 0 BM2 X Y BMW BMH 'INPUT 'PAINT])
```

(BITMAPFROMSTRING

[LAMBDA (STRING FONT BITSERPPIXEL) (* kbr%: "11-Aug-85 16:14")

```
(PROG (BITMAP DS)
(SETQ BITMAP (BITMAPCREATE (STRINGWIDTH STRING FONT)
                          (FONTPROP FONT 'HEIGHT)
                          BITSERPPIXEL))
(SETQ DS (DSPCREATE BITMAP))
(DSPFONT FONT DS)
(MOVETO 0 (FONTPROP FONT 'DESCENT)
        DS)
(PRIN3 STRING DS)
(RETURN BITMAP])
```

(SHADEBITMAP

[LAMBDA (BM T0 T1) (* bas%: "25-APR-82 15:02")
(* Shades bitmap BM with T0 into 0 areas and T1 into 1 areas)

```
(BITBLT NIL NIL NIL BM NIL NIL NIL NIL 'TEXTURE 'INVERT (LOGAND T0 (LOGXOR T0 T1)))
(BITBLT NIL NIL NIL BM NIL NIL NIL NIL 'TEXTURE 'PAINT (LOGAND T0 T1))
(BITBLT NIL NIL NIL BM NIL NIL NIL NIL 'TEXTURE 'ERASE (LOGXOR (LOGOR T0 T1)
65535])
```

```
)
(RPAQ? EDITCOLORMAP.WINDOW NIL)
(DEFINEQ
```

(EDITCOLORMAP

```
[LAMBDA NIL (* kbr%: " 5-Jun-86 22:49")
*)
(* Colormap Editor. Let's user interactively adjust colormap.
```

```
(PROG (XPOS REDREGION GREENREGION BLUERECTION HUERECTION LIGHTNESSREGION SATURATIONREGION BOTTOM)
(COND
  ((NULL EDITCOLORMAP.WINDOW)
   (SETQ EDITCOLORMAP.WINDOW (CREATEW (GETBOXREGION EditColorMapWidth EditColorMapHeight NIL NIL NIL
"Select location of Colormap Editor window.")
"Colormap Editor")))
  (CLRPRMPT)
  (WINDOWPROP EDITCOLORMAP.WINDOW 'BUTTONEVENTFN 'EDITCOLORMAP.BUTTONEVENTFN)
  (WINDOWPROP EDITCOLORMAP.WINDOW 'REPAINTFN 'EDITCOLORMAP.REDISPLAYFN)
  (WINDOWPROP EDITCOLORMAP.WINDOW 'COLOR 0))
  (T (CLEARW EDITCOLORMAP.WINDOW)))
  (REDISPLAYW EDITCOLORMAP.WINDOW])
```

(EDITCOLORMAP.BUTTONEVENTFN

```
[LAMBDA (WINDOW) (* kbr%: " 4-Jun-86 21:21")
```

```
*)
(* Colormap editor. Displays a colormap in a window and allows the user to change it.
```

```
(PROG (REDREGION GREENREGION BLUERECTION HUERECTION LIGHTNESSREGION SATURATIONREGION BOTTOM COLOR COLORMAP
LEVEL LASTX LASTY HLS OLDLEVEL COMPONENT)
(PROGN (SETQ REDREGION (WINDOWPROP WINDOW 'REDREGION))
(SETQ GREENREGION (WINDOWPROP WINDOW 'GREENREGION))
(SETQ BLUERECTION (WINDOWPROP WINDOW 'BLUERECTION))
(SETQ HUERECTION (WINDOWPROP WINDOW 'HUERECTION))
(SETQ LIGHTNESSREGION (WINDOWPROP WINDOW 'LIGHTNESSREGION))
(SETQ SATURATIONREGION (WINDOWPROP WINDOW 'SATURATIONREGION))
(SETQ BOTTOM (fetch (REGION BOTTOM) of REDREGION)))
(SETQ COLOR (WINDOWPROP WINDOW 'COLOR))
(SETQ COLORMAP (SCREENCOLORMAP))
(COND
```

```
  [(LASTMOUSESTATE MIDDLE)
   (COND
    ((NUMBERP (SETQ LEVEL (GETCOLOR#FROMUSER)))
     (WINDOWPROP WINDOW 'COLOR LEVEL)
     (REDISPLAYW WINDOW])
    ((LASTMOUSESTATE LEFT)
```

```
     (SETQ LASTX (LASTMOUSEX WINDOW))
     (SETQ LASTY (LASTMOUSEY WINDOW))
     (COND
      ([SETQ COMPONENT (COND
        ((INSIDEP REDREGION LASTX LASTY)
         'RED)
        ((INSIDEP GREENREGION LASTX LASTY)
         'GREEN)
        ((INSIDEP BLUERECTION LASTX LASTY)
         'BLUE)
        ((INSIDEP HUERECTION LASTX LASTY)
         'HUE)
        ((INSIDEP LIGHTNESSREGION LASTX LASTY)
         'LIGHTNESS)
        ((INSIDEP SATURATIONREGION LASTX LASTY)
         'SATURATION])
       (SETQ OLDLEVEL (WINDOWPROP WINDOW COMPONENT)))
      (until (MOUSESTATE (NOT LEFT)) do
```

```
*)
(* As long as LEFT is down, adjust the color.
```

```
[SETQ LEVEL (IMIN 255 (IMAX 0 (IDIFFERENCE (LASTMOUSEY WINDOW)
)
BOTTOM])
```

```
(COND
  ((NOT (EQ LEVEL OLDLEVEL))
   (CHANGECOLORLEVELS WINDOW COMPONENT LEVEL)
   [SCREENCOLORMAPENTRY COLOR (create RGB
RED _
(WINDOWPROP WINDOW
'RED)
GREEN _
(WINDOWPROP WINDOW
'GREEN)
BLUE _
(WINDOWPROP WINDOW
```

'BLUE]

(SETQ OLDLEVEL LEVEL)]

(EDITCOLORMAP.REDISPLAYFN

[LAMBDA (WINDOW)

(* kbr%: " 4-Jun-86 20:46")

(* Colormap Editor. Let's user interactively adjust colormap.
)

```
(PROG (XPOS REDREGION GREENREGION BLUEREGION HUEREGION LIGHTNESSREGION SATURATIONREGION BOTTOM COLORMAP
      COLOR)
  (CLEARW WINDOW)
  (PROGN (MOVETO 35 4 WINDOW)
    (PRIN1 "RED" WINDOW)
    (SETQ REDREGION '(40 16 10 256))
    (OUTLINEREGION REDREGION 2 NIL WINDOW)
    (WINDOWPROP WINDOW 'REDREGION REDREGION))
  (PROGN (MOVETO 70 4 WINDOW)
    (PRIN1 "GREEN" WINDOW)
    (SETQ GREENREGION '(82 16 10 256))
    (OUTLINEREGION GREENREGION 2 NIL WINDOW)
    (WINDOWPROP WINDOW 'GREENREGION GREENREGION))
  (PROGN (MOVETO 119 4 WINDOW)
    (PRIN1 "BLUE" WINDOW)
    (SETQ BLUEREGION '(128 16 10 256))
    (OUTLINEREGION BLUEREGION 2 NIL WINDOW)
    (WINDOWPROP WINDOW 'BLUEREGION BLUEREGION))
  (PROGN (MOVETO 181 4 WINDOW)
    (PRIN1 "HUE" WINDOW)
    (SETQ HUEREGION '(186 16 10 256))
    (OUTLINEREGION HUEREGION 2 NIL WINDOW)
    (WINDOWPROP WINDOW 'HUEREGION HUEREGION))
  (PROGN (MOVETO 216 4 WINDOW)
    (PRIN1 "LIGHTNESS" WINDOW)
    (SETQ LIGHTNESSREGION '(242 16 10 256))
    (OUTLINEREGION LIGHTNESSREGION 2 NIL WINDOW)
    (WINDOWPROP WINDOW 'LIGHTNESSREGION LIGHTNESSREGION))
  (PROGN (MOVETO 300 4 WINDOW)
    (PRIN1 "SAT" WINDOW)
    (SETQ SATURATIONREGION '(305 16 10 256))
    (OUTLINEREGION SATURATIONREGION 2 NIL WINDOW)
    (WINDOWPROP WINDOW 'SATURATIONREGION SATURATIONREGION))
  (PROGN (SETQ COLORMAP (SCREENCOLORMAP))
    (SETQ COLOR (WINDOWPROP WINDOW 'COLOR))
    (MOVETO 8 250 WINDOW)
    (printout WINDOW .I3 COLOR)
    (DISPLAYCOLORLEVELS WINDOW (ELT COLORMAP COLOR]))
```

(EDITCOLORMAP.VALUELEVEL

[LAMBDA (COMPONENT WINDOWLEVEL)

(* kbr%: " 3-Jun-86 19:55")

(* Value that would be stored in an RGB or HLS corresponding to WINDOWLEVEL.
)

```
(SELECTQ COMPONENT
  (HUE (IQUOTIENT (ITIMES WINDOWLEVEL 360)
    255))
  ((LIGHTNESS SATURATION)
    (FQUOTIENT WINDOWLEVEL 255))
  ((RED GREEN BLUE)
    WINDOWLEVEL)
  (SHOULDNT])
```

(EDITCOLORMAP.WINDOWLEVEL

[LAMBDA (COMPONENT VALUELEVEL)

(* kbr%: " 3-Jun-86 19:55")

(* Given VALUELEVEL of an RGB or HLS, what WINDOWLEVEL should be used to display it? *)

```
(SELECTQ COMPONENT
  (HUE (IQUOTIENT (ITIMES VALUELEVEL 255)
    360))
  ((LIGHTNESS SATURATION)
    (FIX (FTIMES VALUELEVEL 255)))
  ((RED GREEN BLUE)
    VALUELEVEL)
  (SHOULDNT])
```

(CHANGECOLORLEVELS

[LAMBDA (WINDOW COMPONENT WINDOWLEVEL)

(* kbr%: " 3-Jun-86 19:55")

```
(PROG (RGB HLS)
  (DISPLAYCOLORLEVEL WINDOW COMPONENT (EDITCOLORMAP.VALUELEVEL COMPONENT WINDOWLEVEL)
    WINDOWLEVEL)
  (SELECTQ COMPONENT
    ((RED GREEN BLUE)
      [SETQ HLS (RGBTOHLS (WINDOWPROP WINDOW 'RED)
```

```

(WINDOWPROP WINDOW 'GREEN)
(WINDOWPROP WINDOW 'BLUE)
(DISPLAYCOLORLEVEL WINDOW 'HUE (fetch (HLS HUE) of HLS)
 (EDITCOLORMAP.WINDOWLEVEL 'HUE (fetch (HLS HUE) of HLS)))
(DISPLAYCOLORLEVEL WINDOW 'LIGHTNESS (fetch (HLS LIGHTNESS) of HLS)
 (EDITCOLORMAP.WINDOWLEVEL 'LIGHTNESS (fetch (HLS LIGHTNESS) of HLS)))
(DISPLAYCOLORLEVEL WINDOW 'SATURATION (fetch (HLS SATURATION) of HLS)
 (EDITCOLORMAP.WINDOWLEVEL 'SATURATION (fetch (HLS SATURATION) of HLS)))
((HUE LIGHTNESS SATURATION)
 [SETQ RGB (HLSTORGB (EDITCOLORMAP.VALUELEVEL 'HUE (WINDOWPROP WINDOW 'HUE)
 (EDITCOLORMAP.VALUELEVEL 'LIGHTNESS (WINDOWPROP WINDOW 'LIGHTNESS))
 (EDITCOLORMAP.VALUELEVEL 'SATURATION (WINDOWPROP WINDOW 'SATURATION))
 (DISPLAYCOLORLEVEL WINDOW 'RED (fetch (RGB RED) of RGB)
 (fetch (RGB RED) of RGB))
 (DISPLAYCOLORLEVEL WINDOW 'GREEN (fetch (RGB GREEN) of RGB)
 (fetch (RGB GREEN) of RGB))
 (DISPLAYCOLORLEVEL WINDOW 'BLUE (fetch (RGB BLUE) of RGB)
 (fetch (RGB BLUE) of RGB))
(SHOULDNT)])

```

(GETCOLOR#FROMUSER

(* edited%: " 8-SEP-82 21:44")
(* reads a color number from the user.)

```

[LAMBDA NIL
 (PROG (RESPONSE)
 (MOVEV [COND
 ((TYPENAMEP EDIT8BITCOLORMAPNUMBERREADER 'WINDOW)
 EDIT8BITCOLORMAPNUMBERREADER)
 (T (SETQ EDIT8BITCOLORMAPNUMBERREADER (CREATE.NUMBERPAD.READER '(Enter color number to
 edit%:))
 (create POSITION
 XCOORD _ LASTMOUSEX
 YCOORD _ LASTMOUSEY)
 (create POSITION
 XCOORD _ LASTMOUSEX
 YCOORD _ LASTMOUSEY))
 LP (COND
 ([NULL (ERSETQ (SETQ RESPONSE (NUMBERPAD.READ EDIT8BITCOLORMAPNUMBERREADER)
 (* currently there is no way NIL can be returned from NUMBERPAD.READ but there should be a way to quit.)
 (RETURN NIL))
 ((OR (ILESSP RESPONSE 0)
 (IGREATERP RESPONSE 255))
 (PROMPTPRINT "Color numbers must be between 0 and 255.")
 (GO LP))
 (T (RETURN RESPONSE]))

```

(GETCOLOR#FROMSCREEN

(* rrb " 3-NOV-82 13:57")
(* returns the color number of a point selected by the user.)

```

[LAMBDA NIL
 (RESETFORM (CHANGECURSORSSCREEN (COLORSCREENBITMAP))
 (PROG (POS)
 (SETQ POS (GETPOSITION))
 (RETURN (AND POS (BITMAPBIT (COLORSCREENBITMAP)
 (fetch (POSITION XCOORD) of POS)
 (fetch (POSITION YCOORD) of POS))

```

(DISPLAYCOLORLEVEL

(* kbr%: " 4-Jun-86 20:23")

```

[LAMBDA (WINDOW COMPONENT NEWLEVEL WINDOWLEVEL)
 (PROG (REGION)
 (WINDOWPROP WINDOW COMPONENT WINDOWLEVEL)
 (SETQ REGION (SELECTQ COMPONENT
 (RED (WINDOWPROP WINDOW 'REDREGION))
 (BLUE (WINDOWPROP WINDOW 'BLUEREGION))
 (GREEN (WINDOWPROP WINDOW 'GREENREGION))
 (HUE (WINDOWPROP WINDOW 'HUEREGION))
 (LIGHTNESS (WINDOWPROP WINDOW 'LIGHTNESSREGION))
 (SATURATION (WINDOWPROP WINDOW 'SATURATIONREGION))
 (SHOULDNT)))
 [PROGN
 (* Print out new level of COMPONENT.
 *)
 (MOVETO (IDIFFERENCE (fetch (REGION LEFT) of REGION)
 12)
 (IPLUS 8 (fetch (REGION TOP) of REGION))
 WINDOW)
 (* Overstrike extra digits in case the old value was larger.
 *)
 (COND
 ((FIXP NEWLEVEL)
 (printout WINDOW " " .I3 NEWLEVEL))
 (T (printout WINDOW .F5.3 NEWLEVEL))
 (FILLINREGION REGION WINDOWLEVEL GRAYSHADE WINDOW)])

```

(FILLINREGION

```
[LAMBDA (REGION HEIGHT GRAY WINDOW)
  (DSPFILL REGION WHITESHAE 'REPLACE WINDOW)
  (AREAFILL (fetch (REGION LEFT) of REGION)
    (fetch (REGION BOTTOM) of REGION)
    (fetch (REGION WIDTH) of REGION)
    HEIGHT GRAY 'REPLACE WINDOW])
```

(* rrb "23-FEB-82 12:26")
 (* fills part of a region with gray.)

(AREAFILL

```
[LAMBDA (LFT BTM WPTH HGTH SHADE OPERATION WINDOW)
  (BITBLT NIL NIL NIL WINDOW LFT BTM WPTH HGTH 'TEXTURE OPERATION SHADE)]
```

(* fills an area of a window with shade.)

(CENTEREDLEFT

```
[LAMBDA (WIDTH LEFT RIGHT)
  (IQUOTIENT (IDIFFERENCE (IPLUS LEFT RIGHT)
    WIDTH)
  2)]
```

(* rrb "16-FEB-82 14:58")
 (* returns the left point that would leave WIDTH centered
 between LEFT and RIGHT)

(OUTLINEAREA

```
[LAMBDA (LFT BTM WPTH HGHT LINEWIDTH OPERATION WIN)
  (PROG (LEFTPLUSWIDTH RIGHTLINELEFT VERTLINETOP TOPY LINEWIDTH)
    (SETQ LINEWIDTH (OR (NUMBERP LINEWIDTH)
      1))
    (SETQ LFT (IDIFFERENCE LFT LINEWIDTH))
    (SETQ BTM (IDIFFERENCE BTM LINEWIDTH))
    (SETQ WPTH (IPLUS WPTH (ITIMES LINEWIDTH 2)))
    (SETQ HGHT (IPLUS HGHT (ITIMES LINEWIDTH 2)))
    (DRAWLINE LFT BTM LFT (SETQ VERTLINETOP (SUB1 (IPLUS BTM HGHT)))
      LINEWIDTH OPERATION WIN)
    (DRAWLINE (SETQ RIGHTLINELEFT (IDIFFERENCE (IPLUS LFT WPTH)
      LINEWIDTH))
      BTM RIGHTLINELEFT VERTLINETOP LINEWIDTH OPERATION WIN)
    (DRAWLINE (SETQ LEFTPLUSWIDTH (IPLUS LFT LINEWIDTH))
      BTM
      (SETQ RIGHTLINELEFT (SUB1 RIGHTLINELEFT))
      BTM LINEWIDTH OPERATION WIN)
    (DRAWLINE LEFTPLUSWIDTH (SETQ TOPY (ADD1 (IDIFFERENCE VERTLINETOP LINEWIDTH)))
      RIGHTLINELEFT TOPY LINEWIDTH OPERATION WIN])
```

(* rrb "17-FEB-82 10:59")
 (* outlines an area of a window.)

(OUTLINEREGION

```
[LAMBDA (REGION OUTLINEWIDTH OPERATION WIN)
  (OUTLINEAREA (fetch (REGION LEFT) of REGION)
    (fetch (REGION BOTTOM) of REGION)
    (fetch (REGION WIDTH) of REGION)
    (fetch (REGION HEIGHT) of REGION)
    OUTLINEWIDTH OPERATION WIN)]
```

(* rrb "17-FEB-82 10:58")
 (* outlines the region REGION with a width wide line)

)

(DEFINEQ

(ADJUSTCOLORMAP

```
[LAMBDA (PRIMARY DELTA)
  (PROG NIL
    (for COLOR from 0 to (MAXIMUMCOLOR (BITSPERPIXEL (SCREENCOLORMAP)))
      do (COLORLEVEL COLOR PRIMARY (IMIN 255 (IMAX 0 (IPLUS (COLORLEVEL COLOR PRIMARY)
        DELTA))
```

(* kbr%: "5-Jun-86 19:41")
 (* Adds DELTA points of intensity to all values of PRIMARY
 color in SCREENCOLORMAP *)

(SHOWCOLORBLOCKS

```
[LAMBDA (DESTINATION)
  (PROG (BITSPERPIXEL MAXSHADE N WIDTH HEIGHT SHADE)
    (SETQ BITSPERPIXEL (BITSPERPIXEL DESTINATION))
    (SETQ MAXSHADE (MAXIMUMSHADE BITSPERPIXEL))
    [SETQ N (FIXR (SQRT (ADD1 MAXSHADE))
    (SETQ WIDTH (IQUOTIENT (IPLUS (BITMAPWIDTH DESTINATION)
      N -1)
      N))
    (SETQ HEIGHT (IQUOTIENT (IPLUS (BITMAPHEIGHT DESTINATION)
      N -1)
      N))
    (SETQ SHADE 0)
    (for Y from (SUB1 N) to 0 by -1 do (for X from 0 to (SUB1 N) do (BLTSHADE SHADE DESTINATION
      (ITIMES X WIDTH)
      (ITIMES Y HEIGHT)
```

(* kbr%: "17-Aug-85 21:44")
 (* Puts shade blocks onto DESTINATION.
 *)


```

        WIDTH HEIGHT 'REPLACE)
      (SETQ SHADE (ADD1 SHADE))
      (COND
        ((IGREATERP SHADE MAXSHADE)
         (SETQ SHADE 0))

```

(MAPOFACOLOR

```
[LAMBDA (RGB BITSPERPIXEL)
```

(* kbr%: "11-Jul-85 20:04")
 (* creates a gray color map *)

```

  (PROG (MAXCOLOR RED GREEN BLUE OPRED OPGREEN OPBLUE COLORMAP)
    (SETQ MAXCOLOR (MAXIMUMCOLOR BITSPERPIXEL))
    (SETQ RED (fetch (RGB RED) of RGB))
    (SETQ GREEN (fetch (RGB GREEN) of RGB))
    (SETQ BLUE (fetch (RGB BLUE) of RGB))
    (SETQ OPRED (IDIFFERENCE MAXCOLOR RED))
    (SETQ OPGREEN (IDIFFERENCE MAXCOLOR GREEN))
    (SETQ OPBLUE (IDIFFERENCE MAXCOLOR BLUE))
    (SETQ COLORMAP (COLORMAPCREATE (for I from 0 to MAXCOLOR as OPI from MAXCOLOR to 0 by -1
      collect (create RGB
        RED _ (IQUOTIENT (IPLUS (ITIMES OPI OPRED)
          (ITIMES I RED))
          MAXCOLOR)
        GREEN _ (IQUOTIENT (IPLUS (ITIMES OPI OPGREEN)
          (ITIMES I GREEN))
          MAXCOLOR)
        BLUE _ (IQUOTIENT (IPLUS (ITIMES OPI OPBLUE)
          (ITIMES I BLUE))
          MAXCOLOR)))
      BITSPERPIXEL))
    (RETURN COLORMAP])

```

(COLORHEXPATTERN

```
[LAMBDA (LIGHTNESS)
```

(* kbr%: " 3-Jun-86 22:36")
 (* Put a color hex pattern on the color display.
 *)

```

  (PROG (DESTINATION WIDTH HEIGHT BITSPERPIXEL N HEXWIDTH HEXHEIGHT LEFT BOTTOM COLOR MAXI JDIST IDIST)
    (COND
      ((NULL LIGHTNESS)
       (SETQ LIGHTNESS 0.5))
      (SETQ DESTINATION (COLORSCREENBITMAP))
      (SETQ WIDTH (BITMAPWIDTH DESTINATION))
      (SETQ HEIGHT (BITMAPHEIGHT DESTINATION))
      (SETQ BITSPERPIXEL (BITSPERPIXEL DESTINATION))
      (SETQ N (SELECTQ BITSPERPIXEL
        (4 1)
        (8 8)
        (RETURN)))
      (SETQ HEXWIDTH (IQUOTIENT WIDTH (IPLUS (ITIMES 2 N)
        1)))
      (SETQ HEXHEIGHT (IQUOTIENT HEIGHT (IPLUS (ITIMES 2 N)
        1)))
      (BLTSHADE MINIMUMSHADE DESTINATION)
      (SETQ COLOR 0)
      [for J from N to 0 by -1 do (SETQ BOTTOM (ITIMES (IPLUS J N)
        HEXHEIGHT))
        (SETQ MAXI (IDIFFERENCE (IPLUS (ITIMES 2 N)
          1)
          J))
        (for I from 0 to MAXI
          do (SETQ LEFT (IQUOTIENT (ITIMES (IPLUS (ITIMES 2 I)
            J)
            HEXWIDTH)
            2))
            (SETQ COLOR (ADD1 COLOR))
            (BLTSHADE COLOR DESTINATION LEFT BOTTOM HEXWIDTH HEXHEIGHT)
            (SETQ JDIST (FQUOTIENT J N))
            (SETQ IDIST (FDIFFERENCE (FTIMES 2.0 (FQUOTIENT I MAXI))
              1.0))
            (SCREENCOLORMAPENTRY COLOR
              (HLSTORGB (ATAN JDIST IDIST)
                LIGHTNESS
                (SQRT (FQUOTIENT (FPLUS (FTIMES IDIST IDIST)
                  (FTIMES JDIST JDIST))
                  2.0))
              (for J from -1 to (IMINUS N) by -1
                do (SETQ BOTTOM (ITIMES (IPLUS J N)
                  HEXHEIGHT))
                  (SETQ MAXI (IPLUS (IPLUS (ITIMES 2 N)
                    1)
                    J))
                  (for I from 0 to MAXI do (SETQ LEFT (IQUOTIENT (ITIMES (IPLUS (ITIMES 2 I)
                    (IMINUS J))
                    HEXWIDTH)
                    2))
                    (SETQ COLOR (ADD1 COLOR))

```

```

(BLTSHADE COLOR DESTINATION LEFT BOTTOM HEXWIDTH HEXHEIGHT)
(SETQ JDIST (FQUOTIENT J N))
(SETQ IDIST (FDIFFERENCE (FTIMES 2.0 (FQUOTIENT I MAXI))
                          1.0))
(SCREENCOLORMAPENTRY COLOR (HLSTORGB (ATAN JDIST IDIST)
                                     LIGHTNESS
                                     (SQRT (FQUOTIENT (FPLUS (FTIMES IDIST
                                                                IDIST)
                                                                (FTIMES JDIST
                                                                JDIST))
                                             2.0])

```

```

)
(RPAQQ EditColorMapHeight 315)
(RPAQQ EditColorMapWidth 380)
(RPAQQ COLOR#MENUSAVE NIL)
(RPAQQ CONTROLMENUSAVE NIL)
(RPAQQ EDIT8BITCOLORMAPMENU NIL)
(RPAQQ EDIT8BITCOLORMAPNUMBERREADER NIL)
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS COLOR#MENUSAVE CONTROLMENUSAVE EDIT8BITCOLORMAPMENU EDIT8BITCOLORMAPNUMBERREADER EditColorMapHeight
  EditColorMapWidth)
)

```

;;; support for global naming and querying of colors.

(DEFINEQ

(CNSMENUINIT

(* gbn "9-Aug-85 03:11")

```

[LAMBDA NIL
  [SETQ CNSHUEMENU (create MENU
                          ITEMS _ (for I in DICOLOR.hueMapping collect (CAR I))
  [SETQ CNSSATURATIONMENU (create MENU
                          ITEMS _ (for I in DICOLOR.saturationMapping collect (CAR I))
  (SETQ CNSLIGHTNESSMENU (create MENU
                          ITEMS _ (for I in DICOLOR.lightnessMapping collect (CAR I))

```

(CNSTOCSL

(* hdj "12-Apr-85 19:01")

```

[LAMBDA (hue saturation lightness)
  (PROG ((hueAtom (MKATOM hue))
         (saturationAtom (MKATOM saturation))
         (lightnessAtom (MKATOM lightness))
         c s l)
    (if [NOT (SETQ c (fetch (hueRecord ordering) of (ASSOC hueAtom DICOLOR.hueMapping])
                          then (SETQ c DICOLOR.achromatic))
        (if (EQ c DICOLOR.achromatic)
            then (SETQ s DICOLOR.noSaturation)
            else (if [NOT (SETQ s (fetch (saturationRecord ordering) of (ASSOC saturationAtom
                                                                              DICOLOR.saturationMapping])
                                      then (SETQ s DICOLOR.vivid)))
        (SELECTQ hueAtom
          (Black (SETQ l DICOLOR.black))
          (White (SETQ l DICOLOR.white))
          (if [NOT (SETQ l (fetch (lightnessRecord ordering) of (ASSOC lightnessAtom DICOLOR.lightnessMapping])
                                then (SETQ l DICOLOR.medium)))
        (RETURN (LIST c s l])

```

(CNSTORGB

(* hdj "15-Jul-85 12:33")

```

[LAMBDA (saturation lightness hue)
  (LET ((CSL (CNSTOCSL hue saturation lightness))
        (HLSTORGB (APPLY (FUNCTION CSLTOHLS)
                          CSL))

```

(CSLTOCNS

(* hdj "15-Jul-85 12:37")

```

[LAMBDA (c s l)
  (PROG (hue saturation lightness)
    [if (EQ c DICOLOR.achromatic)
      then (SETQ saturation "")
      [SELECTC l
        (DICOLOR.black
          (SETQ hue "Black")
          (SETQ lightness ""))
        (DICOLOR.white
          (SETQ hue "White")
          (SETQ lightness ""))

```


(DICOLOR.lightnessNvalue 1))

2)))

then (RETURN]
(RETURN (LIST c s l])

(CSLTOHLS

(* hdj "15-Jul-85 12:23")

[LAMBDA (c s l)
(PROG (hue saturation lightness)
(if (EQ c DICOLOR.achromatic)
then (SETQ hue 0.0)
(SETQ saturation 0.0)
(SETQ lightness (DICOLOR.lightnessNvalue 1))
else (SETQ hue (DICOLOR.hueNvalue c))
(SETQ saturation (DICOLOR.saturationNvalue s))
(SETQ lightness (DICOLOR.lightnessNvalue 1)))
(RETURN (LIST (MOD (FPLUS 120 (FTIMES hue 360))
360)
lightness saturation])

(RGBTOCNS

(* hdj "15-Jul-85 12:36")

[LAMBDA (Red Green Blue)
(APPLY (FUNCTION CSLTOCNS)
(APPLY (FUNCTION HLSTOCSL)
(RGBTOHLS Red Green Blue])

(RPAQQ DICOLOR.hueMapping

((Achromatic 0.0 -1)
(Red 0.0 0)
(OrangishRed 0.01 1)
(RedOrange 0.02 2)
(ReddishOrange 0.03 3)
(Orange 0.04 4)
(YellowishOrange 0.07 5)
(OrangeYellow 0.1 6)
(OrangishYellow 0.13 7)
(Yellow 0.1673 8)
(GreenishYellow 0.2073 9)
(YellowGreen 0.2473 10)
(YellowishGreen 0.2873 11)
(Green 0.3333 12)
(BluishGreen 0.4133 13)
(GreenBlue 0.4933 14)
(GreenishBlue 0.5733 15)
(Blue 0.6666 16)
(PurplishBlue 0.6816 17)
(BluePurple 0.6966 18)
(BluishPurple 0.7116 19)
(Purple 0.73 20)
(ReddishPurple 0.8 21)
(PurpleRed 0.87 22)
(PurplishRed 0.94 23)
(BrownishRed 0.01 24)
(RedBrown 0.02 25)
(ReddishBrown 0.03 26)
(Brown 0.04 27)
(YellowishBrown 0.07 28)
(BrownYellow 0.1 29)
(BrownishYellow 0.13 30)))

(RPAQQ DICOLOR.lightnessMapping

((Black 0.0 0)
(VeryDark 0.1666 1)
(Dark 0.3333 2)
(Medium 0.5 3)
(Light 0.6666 4)
(VeryLight 0.8333 5)
(White 1.0 6)))

(RPAQQ DICOLOR.saturationMapping

((NoSaturation 0.0 0)
(Grayish 0.25 1)
(Moderate 0.5 2)
(Strong 0.75 3)
(Vivid 1.0 4)))

(RPAQQ NEWCOLORITEM

(New% Color 'CNS "Allows specification of a new color" (SUBITEMS (RGB 'RGB "Specify a new color using Red, Green, Blue sliders")
(CNS 'CNS "Specify a new color using English")))

(RPAQQ? COLORNAMEMENU)

(DEFINEQ

(**DICOLOR.hueN**

[LAMBDA (N) (* hdj "17-Apr-85 13:38")
(**DECLARE** (GLOBALVARS DICOLOR.hueMapping)
(**for** ELT **in** DICOLOR.hueMapping **suchthat** (EQ (**fetch** (hueRecord ordering) **of** ELT)
N])

(**DICOLOR.hueNvalue**

[LAMBDA (N) (* hdj "18-Apr-85 09:58")
(**fetch** (hueRecord value) **of** (**DICOLOR.hueN** N])

(**DICOLOR.hueNname**

[LAMBDA (N) (* hdj "18-Apr-85 10:07")
(**fetch** (hueRecord name) **of** (**DICOLOR.hueN** N])

(**DICOLOR.lightnessN**

[LAMBDA (N) (* hdj "17-Apr-85 13:40")
(**DECLARE** (GLOBALVARS DICOLOR.lightnessMapping)
(**for** ELT **in** DICOLOR.lightnessMapping **suchthat** (EQ (**fetch** (lightnessRecord ordering) **of** ELT)
N])

(**DICOLOR.lightnessNvalue**

[LAMBDA (N) (* hdj "17-Apr-85 13:36")
(**fetch** (lightnessRecord value) **of** (**DICOLOR.lightnessN** N])

(**DICOLOR.lightnessNname**

[LAMBDA (N) (* hdj "17-Apr-85 14:02")
(**fetch** (lightnessRecord name) **of** (**DICOLOR.lightnessN** N])

(**DICOLOR.saturationN**

[LAMBDA (N) (* hdj "17-Apr-85 13:39")
(**DECLARE** (GLOBALVARS DICOLOR.saturationMapping)
(**for** ELT **in** DICOLOR.saturationMapping **suchthat** (EQ (**fetch** (saturationRecord ordering) **of** ELT)
N])

(**DICOLOR.saturationNvalue**

[LAMBDA (N) (* hdj "17-Apr-85 13:36")
(**fetch** (saturationRecord value) **of** (**DICOLOR.saturationN** N])

(**DICOLOR.saturationNname**

[LAMBDA (N) (* hdj "17-Apr-85 14:02")
(**fetch** (saturationRecord name) **of** (**DICOLOR.saturationN** N])

)

(DECLARE%: EVAL@LOAD DONTCOPY

(DECLARE%: EVAL@COMPILE

(RECORD hueRecord (name value ordering))

(RECORD lightnessRecord (name value ordering))

(RECORD saturationRecord (name value ordering))

)

(RPAQQ **DICOLOR.hueConstants**

(DICOLOR.achromatic DICOLOR.blue DICOLOR.bluePurple DICOLOR.bluishGreen DICOLOR.bluishPurple
DICOLOR.brown DICOLOR.brownYellow DICOLOR.brownishRed DICOLOR.brownishYellow DICOLOR.green
DICOLOR.greenBlue DICOLOR.greenishBlue DICOLOR.greenishYellow DICOLOR.orange DICOLOR.orangeYellow
DICOLOR.orangishRed DICOLOR.orangishYellow DICOLOR.purple DICOLOR.purpleRed DICOLOR.purplishBlue
DICOLOR.purplishRed DICOLOR.red DICOLOR.redBrown DICOLOR.redOrange DICOLOR.reddishBrown
DICOLOR.reddishOrange DICOLOR.reddishPurple DICOLOR.yellow DICOLOR.yellowGreen
DICOLOR.yellowishBrown DICOLOR.yellowishGreen DICOLOR.yellowishOrange))

(DECLARE%: EVAL@COMPILE

(RPAQQ **DICOLOR.achromatic** -1)

(RPAQQ **DICOLOR.blue** 16)

(RPAQQ **DICOLOR.bluePurple** 18)

(RPAQQ **DICOLOR.bluishGreen** 13)

(RPAQQ **DICOLOR.bluishPurple** 19)

(RPAQQ **DICOLOR.brown** 27)

- (RPAQQ **DICOLOR.brownYellow** 29)
- (RPAQQ **DICOLOR.brownishRed** 24)
- (RPAQQ **DICOLOR.brownishYellow** 30)
- (RPAQQ **DICOLOR.green** 12)
- (RPAQQ **DICOLOR.greenBlue** 14)
- (RPAQQ **DICOLOR.greenishBlue** 15)
- (RPAQQ **DICOLOR.greenishYellow** 9)
- (RPAQQ **DICOLOR.orange** 4)
- (RPAQQ **DICOLOR.orangeYellow** 6)
- (RPAQQ **DICOLOR.orangishRed** 1)
- (RPAQQ **DICOLOR.orangishYellow** 7)
- (RPAQQ **DICOLOR.purple** 20)
- (RPAQQ **DICOLOR.purpleRed** 22)
- (RPAQQ **DICOLOR.purplishBlue** 17)
- (RPAQQ **DICOLOR.purplishRed** 23)
- (RPAQQ **DICOLOR.red** 0)
- (RPAQQ **DICOLOR.redBrown** 25)
- (RPAQQ **DICOLOR.redOrange** 2)
- (RPAQQ **DICOLOR.reddishBrown** 26)
- (RPAQQ **DICOLOR.reddishOrange** 3)
- (RPAQQ **DICOLOR.reddishPurple** 21)
- (RPAQQ **DICOLOR.yellow** 8)
- (RPAQQ **DICOLOR.yellowGreen** 10)
- (RPAQQ **DICOLOR.yellowishBrown** 28)
- (RPAQQ **DICOLOR.yellowishGreen** 11)
- (RPAQQ **DICOLOR.yellowishOrange** 5)

```
(CONSTANTS DICOLOR.achromatic DICOLOR.blue DICOLOR.bluePurple DICOLOR.bluishGreen DICOLOR.bluishPurple
DICOLOR.brown DICOLOR.brownYellow DICOLOR.brownishRed DICOLOR.brownishYellow DICOLOR.green
DICOLOR.greenBlue DICOLOR.greenishBlue DICOLOR.greenishYellow DICOLOR.orange DICOLOR.orangeYellow
DICOLOR.orangishRed DICOLOR.orangishYellow DICOLOR.purple DICOLOR.purpleRed DICOLOR.purplishBlue
DICOLOR.purplishRed DICOLOR.red DICOLOR.redBrown DICOLOR.redOrange DICOLOR.reddishBrown
DICOLOR.reddishOrange DICOLOR.reddishPurple DICOLOR.yellow DICOLOR.yellowGreen DICOLOR.yellowishBrown
DICOLOR.yellowishGreen DICOLOR.yellowishOrange)
)
```

```
(RPAQQ DICOLOR.saturationConstants (DICOLOR.noSaturation DICOLOR.grayish DICOLOR.moderate DICOLOR.strong
DICOLOR.vivid))
```

```
(DECLARE%: EVAL@COMPILE
```

- (RPAQQ **DICOLOR.noSaturation** 0)
- (RPAQQ **DICOLOR.grayish** 1)
- (RPAQQ **DICOLOR.moderate** 2)
- (RPAQQ **DICOLOR.strong** 3)
- (RPAQQ **DICOLOR.vivid** 4)

```
(CONSTANTS DICOLOR.noSaturation DICOLOR.grayish DICOLOR.moderate DICOLOR.strong DICOLOR.vivid)
)
```

```
(RPAQQ DICOLOR.lightnessConstants (DICOLOR.black DICOLOR.veryDark DICOLOR.dark DICOLOR.medium DICOLOR.light
DICOLOR.veryLight DICOLOR.white))
```

```
(DECLARE%: EVAL@COMPILE
```

- (RPAQQ **DICOLOR.black** 0)

```
{MEDLEY}<obsolete>lispusers>COLOR.;1
```

```
(RPAQQ DICOLOR.veryDark 1)
```

```
(RPAQQ DICOLOR.dark 2)
```

```
(RPAQQ DICOLOR.medium 3)
```

```
(RPAQQ DICOLOR.light 4)
```

```
(RPAQQ DICOLOR.veryLight 5)
```

```
(RPAQQ DICOLOR.white 6)
```

```
(CONSTANTS DICOLOR.black DICOLOR.veryDark DICOLOR.dark DICOLOR.medium DICOLOR.light DICOLOR.veryLight  
DICOLOR.white)
```

```
)  
)
```

```
(CNSMENUINIT)
```

```
(FILESLOAD LLCOLOR READNUMBER)
```

```
(SETQ EDITBMMENU NIL)
```

```
(MOVD 'ARRAYP 'COLORMAPP)
```

```
(PUTPROPS COLOR COPYRIGHT ("Xerox Corporation" 1982 1983 1985 1986 1987))
```

FUNCTION INDEX

ADJUSTCOLORMAP	8	DICOLOR.lightnessNname	13	GETCOLOR#FROMUSER	7
AREAFILL	8	DICOLOR.lightnessNvalue	13	HLSLEVEL	2
BITMAPFROMSTRING	4	DICOLOR.saturationN	13	HLSTOCSL	11
CENTEREDLEFT	8	DICOLOR.saturationNname	13	HLSTORGB	2
CHANGECOLORLEVELS	6	DICOLOR.saturationNvalue	13	HLSVALUEFN	2
CNSMENUINIT	10	DISPLAYCOLORLEVEL	7	HLSVALUEFROMLEVEL	2
CNSTOCSL	10	DISPLAYCOLORLEVELS	1	LEVELFROMHLSVALUE	3
CNSTORGB	10	DISPLAYHLSLEVELS	2	MAPOFACOLOR	9
COLORHEXPATTERN	9	EDITCOLORMAP	5	OUTLINEAREA	8
CSLTOCNS	10	EDITCOLORMAP.BUTTONEVENTFN	5	OUTLINEREGION	8
CSLTOHLS	12	EDITCOLORMAP.REDISPLAYFN	6	OVERPAINT	4
DICOLOR.FROM.USER	11	EDITCOLORMAP.VALUELEVEL	6	RAINBOWMAP	3
DICOLOR.hueN	13	EDITCOLORMAP.WINDOWLEVEL	6	RGBTOCNS	12
DICOLOR.hueNname	13	FILLINREGION	7	RGBTOHLS	3
DICOLOR.hueNvalue	13	GETCNS	11	SHADEBITMAP	4
DICOLOR.lightnessN	13	GETCOLOR#FROMSCREEN	7	SHOWCOLORBLOCKS	8

CONSTANT INDEX

DICOLOR.achromatic	14	DICOLOR.grayish	14	DICOLOR.orangishRed	14	DICOLOR.redOrange	14
DICOLOR.black	15	DICOLOR.green	14	DICOLOR.orangishYellow	14	DICOLOR.strong	14
DICOLOR.blue	14	DICOLOR.greenBlue	14	DICOLOR.purple	14	DICOLOR.veryDark	15
DICOLOR.bluePurple	14	DICOLOR.greenishBlue	14	DICOLOR.purpleRed	14	DICOLOR.veryLight	15
DICOLOR.bluishGreen	14	DICOLOR.greenishYellow	14	DICOLOR.purplishBlue	14	DICOLOR.vivid	14
DICOLOR.bluishPurple	14	DICOLOR.light	15	DICOLOR.purplishRed	14	DICOLOR.white	15
DICOLOR.brown	14	DICOLOR.medium	15	DICOLOR.red	14	DICOLOR.yellow	14
DICOLOR.brownishRed	14	DICOLOR.moderate	14	DICOLOR.redBrown	14	DICOLOR.yellowGreen	14
DICOLOR.brownishYellow	14	DICOLOR.noSaturation	14	DICOLOR.reddishBrown	14	DICOLOR.yellowishBrown	14
DICOLOR.brownYellow	14	DICOLOR.orange	14	DICOLOR.reddishOrange	14	DICOLOR.yellowishGreen	14
DICOLOR.dark	15	DICOLOR.orangeYellow	14	DICOLOR.reddishPurple	14	DICOLOR.yellowishOrange	14

VARIABLE INDEX

COLOR#MENUSAVE	10	DICOLOR.lightnessConstants	14	EDIT8BITCOLORMAPNUMBERREADER	10
COLORNAMEMENU	12	DICOLOR.lightnessMapping	12	EDITCOLORMAP.WINDOW	5
CONTROLMENUSAVE	10	DICOLOR.saturationConstants	14	EditColorMapHeight	10
DICOLOR.hueConstants	13	DICOLOR.saturationMapping	12	EditColorMapWidth	10
DICOLOR.hueMapping	12	EDIT8BITCOLORMAPMENU	10	NEWCOLORITEM	12

RECORD INDEX

hueRecord	13	lightnessRecord	13	saturationRecord	13
-----------------	----	-----------------------	----	------------------------	----