File created: 15-Jun-90 13:49:37 **{DSK}<usr>local>lde>lispcore>internal>library>DICOLOR.;2**

changes to: (VARS DICOLORCOMS)

previous date: 15-Aug-85 19:44:58 {DSK}<usr>local>lde>lispcore>internal>library>DICOLOR.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

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

(RPAQQ DICOLORCOMS
       ((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%: DONTCOPY (*)
               (RECORDS hueRecord lightnessRecord saturationRecord)
               (CONSTANTS * DICOLOR.hueConstants)
               (CONSTANTS * DICOLOR.saturationConstants)
               (CONSTANTS * DICOLOR.lightnessConstants))))

(DEFINEQ

(CNSMENUINIT
  [LAMBDA NIL                                                    (* gbn " 9-Aug-85 03:11")
    [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
  [LAMBDA (hue saturation lightness)                             (* hdj "12-Apr-85 19:01")
    (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
  [LAMBDA (saturation lightness hue)                             (* hdj "15-Jul-85 12:33")
    (LET ((CSL (CNSTOCSL hue saturation lightness)))
         (HLSTORGB (APPLY (FUNCTION CSLTOHLS)
                          CSL])

(CSLTOCNS
  [LAMBDA (c s l)                                                (* hdj "15-Jul-85 12:37")
    (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 ""))
                        (PROGN (SETQ hue "Gray")
                               (SETQ lightness (MKSTRING (fetch (lightnessRecord name) of (DICOLOR.lightnessN l]
            else (SETQ hue (fetch (hueRecord name) of (DICOLOR.hueN c)))
                 (SETQ saturation (fetch (saturationRecord name) of (DICOLOR.saturationN s)))
                 (SETQ lightness (fetch (lightnessRecord name) of (DICOLOR.lightnessN l]
          (RETURN (LIST saturation lightness hue])
```

(DICOLOR.FROM.USER

```
[LAMBDA (NAMES?)                                                    (* gbn " 9-Aug-85 04:51")

        (* * returns an RGB triple. If NAMES? prompts the user first with the global color name menu.
        She can then choose NEWCOLOR which can be specified as RGB or CNS)

   (PROG (NAME RGB)
         (if NAMES?
             then                                                   (* first try to get a color name)
                  [SETQ NAME (MENU (OR COLORNAMEMENU (SETQ COLORNAMEMENU
                                                           (CREATE MENU
                                                                   ITEMS _ (CONS NEWCOLORITEM
                                                                                 (FOR ENTRY IN COLORNAMES
                                                                                    COLLECT (CAR ENTRY]
                  (if (NOT NAME)
                      then                                          (* the user clicked outside the menu)
                           (RETURN))
                  [SETQ RGB (SELECTQ NAME
                                 (RGB (READCOLOR1 "specify new color"))
                                 (CNS (APPLY (FUNCTION CNSTORGB)
                                             (GETCNS)))
                                 (RETURN (CDR (ASSOC NAME COLORNAMES]
                  (if (NOT (SETQ NAME (TTYIN "New color name? ")))
                      then                                          (* user must have decided that she didn't want to keep
                                                                    (name) the color)
                           (RETURN))
                  (push COLORNAMES (CONS (CAR NAME)
                                         RGB))
                  (SETQ COLORNAMEMENU NIL)                          (* invalidate the menu)
                  (RETURN RGB])
```

(GETCNS

```
[LAMBDA NIL                                                         (* gbn " 9-Aug-85 03:13")
  (LIST (MENU CNSLIGHTNESSMENU)
        (MENU CNSSATURATIONMENU)
        (MENU CNSHUEMENU])
```

(HLSTOCSL

```
[LAMBDA (hue lightness saturation)                                  (* hdj "15-Jul-85 12:14")
  (LET ((ISLHue (FQUOTIENT (MOD (PLUS hue 240)
                                360)
                           360)))
       (PROG (c s l)
             (for old s from DICOLOR.noSaturation to DICOLOR.vivid
                do (if (EQ s DICOLOR.vivid)
                       then (RETURN))
                   (if (LEQ saturation (PLUS (DICOLOR.saturationNvalue s)
                                             (QUOTIENT (DIFFERENCE (DICOLOR.saturationNvalue (ADD1 s))
                                                                   (DICOLOR.saturationNvalue s))
                                                       2)))
                       then (RETURN)))
             [if (EQ s DICOLOR.noSaturation)
                 then (SETQ c DICOLOR.achromatic)
                      (for old l from DICOLOR.black to DICOLOR.white
                         do (if (EQ l DICOLOR.white)
                                then (RETURN))
                            (if (LEQ lightness (PLUS (DICOLOR.lightnessNvalue l)
                                                     (QUOTIENT (DIFFERENCE (DICOLOR.lightnessNvalue (ADD1 l))
                                                                           (DICOLOR.lightnessNvalue l))
                                                               2)))
                                then (RETURN)))
               else (for old c from DICOLOR.red to DICOLOR.purplishRed
                       do                                           (* (HELP c))
                          (if (EQ c DICOLOR.purplishRed)
                              then (if (GREATERP ISLHue (PLUS (DICOLOR.hueNvalue c)
                                                              (QUOTIENT (DIFFERENCE 1 (DICOLOR.hueNvalue c))
                                                                        2)))
                                       then (SETQ c DICOLOR.red))
                                   (RETURN))
                          (if (LEQ ISLHue (PLUS (DICOLOR.hueNvalue c)
                                                (QUOTIENT (DIFFERENCE (DICOLOR.hueNvalue (ADD1 c))
                                                                      (DICOLOR.hueNvalue c))
                                                          2)))
                              then (RETURN)))
                       (for old l from DICOLOR.veryDark to DICOLOR.veryLight
                          do (if (EQ l DICOLOR.veryLight)
                                 then (RETURN))
                             (if (LEQ lightness (PLUS (DICOLOR.lightnessNvalue l)
                                                      (QUOTIENT (DIFFERENCE (DICOLOR.lightnessNvalue (ADD1 l))
                                                                            (DICOLOR.lightnessNvalue l))
                                                                2)))
                                 then (RETURN]
             (RETURN (LIST c s l])
```

(CSLTOHLS

```
  [LAMBDA (c s l)                                          (* hdj "15-Jul-85 12:23")
    (PROG (hue saturation lightness)
          (if (EQ c DICOLOR.achromatic)
              then (SETQ hue 0.0)
                   (SETQ saturation 0.0)
                   (SETQ lightness (DICOLOR.lightnessNvalue l))
            else (SETQ hue (DICOLOR.hueNvalue c))
                 (SETQ saturation (DICOLOR.saturationNvalue s))
                 (SETQ lightness (DICOLOR.lightnessNvalue l)))
          (RETURN (LIST (MOD (FPLUS 120 (FTIMES hue 360))
                             360)
                        lightness saturation])
```

(RGBTOCNS

```
  [LAMBDA (Red Green Blue)                                 (* hdj "15-Jul-85 12:36")
    (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")))
)

(RPAQ? 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%: 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)

(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)
)
)

(PUTPROPS DICOLOR COPYRIGHT ("Venue & Xerox Corporation" 1985 1990))
```

FUNCTION INDEX

CONSTANT INDEX

VARIABLE INDEX

RECORD INDEX