

File created: 26-Nov-2023 09:46:44 {WMEDLEY}<library>HRULE.;5

edit by: rmk

changes to: (FNS HRULE.CREATE VRULE.CREATE CROPMARK.CREATE)

previous date: 25-Jul-2022 15:09:26 {WMEDLEY}<library>HRULE.;4

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ **HRULECOMS**

[[COMS
;; Horizontal rules

```
(FNS HRULE.CREATE HRULE.DISPLAYFN HRULE.GETFN HRULE.IMAGEBOXFN HRULE.PUTFN HRULE.COPYFN
  HRULE.WHENOPERATEDONFN)
(INITVARS (HRULE.DEFAULT.WIDTH 2))
(VARS (HRULEFNS ' (HRULE.CREATE HRULE.DISPLAYFN HRULE.GETFN HRULE.IMAGEBOXFN HRULE.PUTFN
  HRULE.COPYFN HRULE.WHENOPERATEDONFN)
  (HRULE.IMAGEFNS (IMAGEFNSCREATE (FUNCTION HRULE.DISPLAYFN)
    (FUNCTION HRULE.IMAGEBOXFN)
    (FUNCTION HRULE.PUTFN)
    (FUNCTION HRULE.GETFN)
    (FUNCTION HRULE.COPYFN)
    (FUNCTION NIL)
    (FUNCTION NIL)
    (FUNCTION NIL)
    (FUNCTION NIL)
    (FUNCTION NIL)
    (FUNCTION NIL)
    (FUNCTION HRULE.WHENOPERATEDONFN)
    (FUNCTION (LAMBDA (OBJ)
      (CONCAT (CHARACTER (CHARCODE EOL))
        (ALLOCSTRING 20 "-")
        (CHARACTER (CHARCODE EOL))
```

[COMS
;; Vertical rules

```
(FNS VRULE.CREATE VRULE.DISPLAYFN VRULE.GETFN VRULE.GETFN2 VRULE.IMAGEBOXFN VRULE.PUTFN
  VRULE.COPYFN VRULE.WHENOPERATEDONFN)
(INITVARS (VRULE.DEFAULT.HEIGHT 12))
(VARS (VRULE.IMAGEFNS (IMAGEFNSCREATE (FUNCTION VRULE.DISPLAYFN)
  (FUNCTION VRULE.IMAGEBOXFN)
  (FUNCTION VRULE.PUTFN)
  (FUNCTION VRULE.GETFN2)
  (FUNCTION VRULE.COPYFN)
  (FUNCTION NIL)
  (FUNCTION NIL)
  (FUNCTION NIL)
  (FUNCTION NIL)
  (FUNCTION NIL)
  (FUNCTION NIL)
  (FUNCTION VRULE.WHENOPERATEDONFN)
  (FUNCTION NIL]
```

;; Old fixed-width-rule reader:

```
(ADDVARS (IMAGEOBJGETFNS (VRULE.GETFN])
```

(COMS
;; Cropping marks

```
(FNS CROPMARK.CREATE CROPMARK.DISPLAYFN CROPMARK.GETFN CROPMARK.IMAGEBOXFN CROPMARK.PUTFN
  CROPMARK.COPYFN CROPMARK.WHENOPERATEDONFN)
(BITMAPS CROPMARK.IMAGE)
(INITVARS (CROPMARK.DEFAULT.PAGESIZE (LIST 612 792)))
(VARS (CROPMARK.IMAGEFNS (IMAGEFNSCREATE (FUNCTION CROPMARK.DISPLAYFN)
  (FUNCTION CROPMARK.IMAGEBOXFN)
  (FUNCTION CROPMARK.PUTFN)
  (FUNCTION CROPMARK.GETFN)
  (FUNCTION CROPMARK.COPYFN)
  (FUNCTION NIL)
  (FUNCTION NIL)
  (FUNCTION NIL)
  (FUNCTION NIL)
  (FUNCTION NIL)
  (FUNCTION NIL)
  (FUNCTION CROPMARK.WHENOPERATEDONFN)
  (FUNCTION NIL])
```

;; Horizontal rules

(DEFINEQ

(HRULE.CREATE

[LAMBDA (WIDTH)

; Edited 26-Nov-2023 09:45 by rmk
(* jds "11-Sep-85 16:36")

;;; Create a Horizontal-Rule image object. WIDTH may be NIL to default, a number, for a single rule with its width in points (and fractions thereof), or a list of alternating black and white widths. E.g., to get a hairline over 1pt white over 3pt rule, specify (0.5 1 3)

```
(PROG ((HRULE (IMAGEOBJCREATE NIL HRULE.IMAGEFNS)))
  (COND
    (NOT WIDTH) ; Use the default width
    (IMAGEOBJPROP HRULE 'RULE.WIDTH HRULE.DEFAULT.WIDTH)
    (RETURN HRULE))
  (NUMBERP WIDTH)
  (IMAGEOBJPROP HRULE 'RULE.WIDTH WIDTH)
  (RETURN HRULE))
  (AND (LISTP WIDTH)
    (EVERY WIDTH (FUNCTION NUMBERP))) ; It's a list of numbers. Add (QUOTE em) up
  (IMAGEOBJPROP HRULE 'RULE.WIDTH WIDTH)
  (RETURN HRULE))
  (T ; Something was specified, and there was a non-number in it...
    (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT "HRULE with non-numeric width: " WIDTH)
      T]))
```

(HRULE.DISPLAYFN

[LAMBDA (HRULE IMAGE.STREAM) (* jds "17-Oct-85 11:35")

(* function which displays the bitmap of the hrule on the display or calls an {inter}press function to draw the rule on a file)

```
(LET* ((RULEWIDTH (IMAGEOBJPROP HRULE 'RULE.WIDTH))
  (WIDTHS (COND
    ((LISTP RULEWIDTH)
      (REVERSE RULEWIDTH))
    (T RULEWIDTH)))
  (SCALE (DSPSCALE NIL IMAGE.STREAM))
  (X (DSPXPOSITION NIL IMAGE.STREAM))
  (Y (DSPLYPOSITION NIL IMAGE.STREAM)))
  (bind [RULING _ (OR (NLISTP WIDTHS)
    (ODDP (FLENGTH WIDTHS)]
    for THICKNESS inside WIDTHS do
```

(* * Run thru the list of alternating rules and spaces %. Display the rules, and skip over the spaces)

```
[SETQ WIDTH (IMAX 1 (FIXR (FTIMES SCALE THICKNESS)
  (* Compute the width of this piece, in stream units.)
[COND
  (RULING (* If we're supposed to be drawing, draw the line)
    (SELECTQ (IMAGESTREAMTYPE IMAGE.STREAM)
      (DISPLAY (BITBLT NIL 0 0 IMAGE.STREAM X Y
        (fetch XSIZE of (IMAGEOBJPROP
          HRULE
            'BOUNDBOX))
          WIDTH
            'TEXTURE
            'PAINT BLACKSHADE))
      (DRAWLINE X (IPLUS Y (LRSH WIDTH 1))
        [IPLUS X (fetch XSIZE of (IMAGEOBJPROP
          HRULE
            'BOUNDBOX)]
          (IPLUS Y (LRSH WIDTH 1))
            WIDTH
            'PAINT IMAGE.STREAM]
      (add Y WIDTH)
    (SETQ RULING (NOT RULING])
```

(HRULE.GETFN

[LAMBDA (INPUT.STREAM TEXTSTREAM) (* gbn "10-Jan-85 02:56")
(* reads the width and creates an HRULE)

```
(HRULE.CREATE (READ INPUT.STREAM])
```

(HRULE.IMAGEBOXFN

[LAMBDA (HRULE IMAGE.STREAM CURRENT.X RIGHT.MARGIN) (* jds "11-Sep-85 17:12")
(* returns an imagebox describing the size of the scaled bitmap.
without caching)

```
(LET [(SCALE (DSPSCALE NIL IMAGE.STREAM))
  (WIDTHS (IMAGEOBJPROP HRULE 'RULE.WIDTH))
  (create IMAGEBOX
    XSIZE _ (IMAX (IDIFFERENCE RIGHT.MARGIN CURRENT.X)
      0)
    YSIZE _ [for THICKNESS inside WIDTHS sum (IMAX 1 (FIXR (FTIMES SCALE THICKNESS)
      YDESC _ 0
      XKERN _ 0])
```

(HRULE.PUTFN

[LAMBDA (HRULE OUTPUT.STREAM) (* gbn "13-May-84 14:21")
(* prints only the rule.width to the file, the rest can be discovered)

```
(PRINT (IMAGEOBJPROP HRULE 'RULE.WIDTH)
  OUTPUT.STREAM])
```

(HRULE.COPYFN

[LAMBDA (IMAGEOBJ FROMSTREAM TOSTREAM)

(* jds "22-Feb-85 13:56")
(* This function does not build the bitmap but lets the
imageboxfn cache a bitmap)

(HRULE.CREATE (IMAGEOBJPROP IMAGEOBJ 'RULE.WIDTH)
TOSTREAM])

(HRULE.WHENOPERATEDONFN

[LAMBDA (A B C C)

(* gbn "6-Jan-85 13:23")
(* DUMMY)

)

(RPAQ? HRULE.DEFAULT.WIDTH 2)

(RPAQQ HRULEFN (HRULE.CREATE HRULE.DISPLAYFN HRULE.GETFN HRULE.IMAGEBOXFN HRULE.PUTFN HRULE.COPYFN
HRULE.WHENOPERATEDONFN)

(RPAQ HRULE.IMAGEFN

[IMAGEFNCREATE (FUNCTION HRULE.DISPLAYFN)
(FUNCTION HRULE.IMAGEBOXFN)
(FUNCTION HRULE.PUTFN)
(FUNCTION HRULE.GETFN)
(FUNCTION HRULE.COPYFN)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION HRULE.WHENOPERATEDONFN)
(FUNCTION (LAMBDA (OBJ)
(CONCAT (CHARACTER (CHARCODE EOL))
(ALLOCSTRING 20 "-")
(CHARACTER (CHARCODE EOL))

:: Vertical rules

(DEFINEQ

(VRULE.CREATE

[LAMBDA (WIDTH HEIGHT DASHING)

; Edited 26-Nov-2023 09:45 by rmk
; Edited 8-Oct-92 16:46 by sybalsky:mv:envos

:: Create a Vertical-Rule image object. HEIGHT may be NIL to default, a number, for a single rule with its width in points (and fractions thereof), or
:: a list of alternating black and white widths. E.g., to get a hairline over 1pt white over 3pt rule, specify (0.5 1 3)

(PROG ((VRULE (IMAGEOBJCREATE NIL VRULE.IMAGEFN)))
(COND
((NOT WIDTH) ; Use the default width
(IMAGEOBJPROP VRULE 'RULE.WIDTH 0.5)
(NUMBERP WIDTH)
(IMAGEOBJPROP VRULE 'RULE.WIDTH WIDTH))
((AND (LISTP WIDTH)
(EVERY WIDTH (FUNCTION NUMBERP))) ; It's a list of numbers. Add 'em up
(IMAGEOBJPROP VRULE 'RULE.WIDTH WIDTH))
(COND
((NOT HEIGHT) ; Use the default width
(IMAGEOBJPROP VRULE 'RULE.HEIGHT VRULE.DEFAULT.HEIGHT)
(RETURN VRULE))
(NUMBERP HEIGHT)
(IMAGEOBJPROP VRULE 'RULE.HEIGHT HEIGHT)
(RETURN VRULE))
((AND (LISTP HEIGHT)
(EVERY HEIGHT (FUNCTION NUMBERP))) ; It's a list of numbers. Add 'em up
(IMAGEOBJPROP VRULE 'RULE.HEIGHT HEIGHT)
(RETURN VRULE))
(T ; Something was specified, and there was a non-number in it...
(TEDIT.PROMPTPRINT TEXTOBJ (CONCAT "VRULE with non-numeric height: " HEIGHT)
T)))
(IMAGEOBJPROP VRULE 'RULE.DASHING DASHING])

(VRULE.DISPLAYFN

[LAMBDA (HRULE IMAGE.STREAM)

; Edited 29-Sep-92 21:01 by jds

:: function which displays the bitmap of the hrule on the display or calls an {inter}press function to draw the rule on a file

(LET* ((RULEHEIGHT (IMAGEOBJPROP HRULE 'RULE.HEIGHT))
(WIDTHS (OR (IMAGEOBJPROP HRULE 'RULE.WIDTH)
0.5))
(SCALE (DSPSCALE NIL IMAGE.STREAM))
(REALHEIGHT (FIXR (FTIMES SCALE RULEHEIGHT)))
(X (DSPXPOSITION NIL IMAGE.STREAM))
(Y (DSPYPOSITION NIL IMAGE.STREAM)))

```
(bind [RULING _ (OR (NLISTP WIDTHS)
                   (ODDP (FLENGTH WIDTHS)]
      WIDTH for THICKNESS inside WIDTHS
do
```

;;; Run thru the list of alternating rules and spaces . Display the rules, and skip over the spaces

```
[SETQ WIDTH (IMAX 1 (FIXR (FTIMES SCALE THICKNESS)
                          ; Compute the width of this piece, in stream units.
[COND
  (RULING
   (SELECTQ (IMAGESTREAMTYPE IMAGE.STREAM)
             ; If we're supposed to be drawing, draw the line
             (DISPLAY (BITBLT NIL 0 0 IMAGE.STREAM X Y 1 6 'TEXTURE 'PAINT BLACKSHADE))
             (DRAWLINE (IPLUS X (LRSH WIDTH 1))
                       Y
                       (IPLUS X (LRSH WIDTH 1))
                       (IDIFFERENCE Y (CL:IF (>= REALHEIGHT 0)
                                             (IMAX 1 REALHEIGHT)
                                             (IMIN -1 REALHEIGHT))))
             WIDTH
             'PAINT IMAGE.STREAM]
  (add X WIDTH)
  (SETQ RULING (NOT RULING])
```

(VRULE.GETFN

```
[LAMBDA (INPUT.STREAM TEXTSTREAM)
  ;; reads the width and creates a VRULE
  (VRULE.CREATE 0.5 (READ INPUT.STREAM])
```

; Edited 15-May-91 13:20 by jds

(VRULE.GETFN2

```
[LAMBDA (INPUT.STREAM TEXTSTREAM)
  ;; reads the width and creates a VRULE
  (VRULE.CREATE (READ INPUT.STREAM)
                (READ INPUT.STREAM)
                (READ INPUT.STREAM])
```

; Edited 8-Oct-92 16:46 by sybalsky:mv:envos

(VRULE.IMAGEBOXFN

```
[LAMBDA (HRULE IMAGE.STREAM CURRENT.X RIGHT.MARGIN)
  ;; returns an imagebox describing the size of the scaled bitmap. without caching
  (LET ((SCALE (DSPSCALE NIL IMAGE.STREAM))
        (WIDTHS 0.5))
    (create IMAGEBOX
            XSIZE _ 0
            YSIZE _ 1
            YDESC _ 0
            XKERN _ 0])
```

; Edited 30-Apr-91 00:06 by jds

(VRULE.PUTFN

```
[LAMBDA (HRULE OUTPUT.STREAM)
  ;; prints WIDTH, HEIGHT and DASHING to the file.
  (PRINT (IMAGEOBJPROP HRULE 'RULE.WIDTH)
         OUTPUT.STREAM)
  (PRINT (IMAGEOBJPROP HRULE 'RULE.HEIGHT)
         OUTPUT.STREAM)
  (PRINT (IMAGEOBJPROP HRULE 'RULE.DASHING)
         OUTPUT.STREAM])
```

; Edited 6-Jul-92 07:02 by jds

(VRULE.COPYFN

```
[LAMBDA (IMAGEOBJ FROMSTREAM TOSTREAM)
  (VRULE.CREATE (IMAGEOBJPROP IMAGEOBJ 'RULE.WIDTH)
                (IMAGEOBJPROP IMAGEOBJ 'RULE.HEIGHT])
```

; Edited 5-Jul-92 17:03 by jds
; This function does not build the bitmap but lets the imageboxfn
; cache a bitmap

(VRULE.WHENOPERATEDONFN

```
[LAMBDA (A B C C)
  ])
```

(* gbn " 6-Jan-85 13:23")
(* DUMMY)

(RPAQ? VRULE.DEFAULT.HEIGHT 12)

(RPAQ VRULE.IMAGEFN
(IMAGEFNCREATE (FUNCTION VRULE.DISPLAYFN)
(FUNCTION VRULE.IMAGEBOXFN)
(FUNCTION VRULE.PUTFN)

```
(FUNCTION VRULE.GETFN2)
(FUNCTION VRULE.COPYFN)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION VRULE.WHENOPERATEDONFN)
(FUNCTION NIL))
```

:: Old fixed-width-rule reader:

```
(ADDTOVAR IMAGEOBJGETFNS (VRULE.GETFN))
```

:: Cropping marks

```
(DEFINEQ
```

(CROPMARK.CREATE

```
[LAMBDA (WIDTH HEIGHT) ; Edited 26-Nov-2023 09:46 by rmk
; Edited 5-Jun-91 14:56 by jds
```

:: Create a CROPMARK, that prints crop-marks for a page that is WIDTH points wide and HEIGHT points high.

```
(PROG ((CROPMARK (IMAGEOBJCREATE NIL CROPMARK.IMAGEFNS))
(COND
  (NOT HEIGHT) ; Use the default width
  (IMAGEOBJPROP CROPMARK 'PAGE.SIZE CROPMARK.DEFAULT.PAGESIZE)
  (RETURN CROPMARK))
  (NUMBERP HEIGHT)
  (IMAGEOBJPROP CROPMARK 'PAGE.SIZE (LIST WIDTH HEIGHT))
  (RETURN CROPMARK))
  (AND (LISTP HEIGHT)
  (EVERY HEIGHT (FUNCTION NUMBERP))) ; It's a list of numbers. Add 'em up
  (IMAGEOBJPROP CROPMARK 'PAGE.SIZE (LIST WIDTH HEIGHT))
  (RETURN CROPMARK))
  (T ; Something was specified, and there was a non-number in it...
  (TEDIT.PROMPTPRINT TEXTOBJ (CONCAT "CROPMARK with non-numeric height: " HEIGHT)
  T]))
```

(CROPMARK.DISPLAYFN

```
[LAMBDA (CROPMARK IMAGE.STREAM) ; Edited 5-Jun-91 15:05 by jds
```

:: function which displays the bitmap of the hrule on the display or calls an {inter}press function to draw the rule on a file

```
(LET* [(PAGESIZE (IMAGEOBJPROP CROPMARK 'PAGE.SIZE))
(WIDTH (CAR PAGESIZE))
(HEIGHT (CADR PAGESIZE))
(SCALE (DSPSCALE NIL IMAGE.STREAM))
(THICK (IMAX 1 (FIXR (FTIMES SCALE 0.5))
(SELECTQ (IMAGESTREAMTYPE IMAGE.STREAM)
  (DISPLAY (BITBLT CROPMARK.IMAGE 0 0 IMAGE.STREAM (DSPXPOSITION NIL IMAGE.STREAM)
  (DSPYPOSITION NIL IMAGE.STREAM)
  9 9 'INPUT 'PAINT)))
  (PROGN (DRAWLINE -12 0 0 0 THICK 'PAINT IMAGE.STREAM)
  (DRAWLINE 0 -12 0 0 THICK 'PAINT IMAGE.STREAM)
  (DRAWLINE -12 HEIGHT 0 HEIGHT THICK 'PAINT IMAGE.STREAM)
  (DRAWLINE 0 (+ 12 HEIGHT)
  0 HEIGHT THICK 'PAINT IMAGE.STREAM)
  (DRAWLINE (+ WIDTH 12)
  0 WIDTH 0 THICK 'PAINT IMAGE.STREAM)
  (DRAWLINE WIDTH -12 WIDTH 0 THICK 'PAINT IMAGE.STREAM)
  (DRAWLINE (+ WIDTH 12)
  HEIGHT WIDTH HEIGHT THICK 'PAINT IMAGE.STREAM)
  (DRAWLINE WIDTH (+ 12 HEIGHT)
  WIDTH HEIGHT THICK 'PAINT IMAGE.STREAM)])
```

(CROPMARK.GETFN

```
[LAMBDA (INPUT.STREAM TEXTSTREAM) ; Edited 5-Jun-91 15:06 by jds
```

:: reads the width and creates a VRULE

```
(LET ((PAGESIZE (READ INPUT.STREAM)))
  (VRULE.CREATE (CAR PAGESIZE)
  (CADR PAGESIZE]))
```

(CROPMARK.IMAGEBOXFN

```
[LAMBDA (HRULE IMAGE.STREAM CURRENT.X RIGHT.MARGIN) ; Edited 5-Jun-91 15:07 by jds
```

:: returns an imagebox describing the size of the scaled bitmap. without caching

```
(LET ((SCALE (DSPSCALE NIL IMAGE.STREAM))
(WIDTHS 0.5))
  (SELECTQ (IMAGESTREAMTYPE IMAGE.STREAM)
  (DISPLAY (create IMAGEBOX
  XSIZE _ 9
```

```

                YSIZE _ 9
                YDESC _ 0
                XKERN _ 0))
(create IMAGEBOX
  XSIZE _ 0
  YSIZE _ 0
  YDESC _ 0
  XKERN _ 0])

```

(CROPMARK.PUTFN

```

[LAMBDA (HRULE OUTPUT.STREAM)
;; prints only the rule.width to the file, the rest can be discovered
(PRINT (IMAGEOBJPROP HRULE 'PAGE.SIZE)
OUTPUT.STREAM])

```

; Edited 5-Jun-91 15:08 by jds

(CROPMARK.COPYFN

```

[LAMBDA (IMAGEOBJ FROMSTREAM TOSTREAM)

(CROPMARK.CREATE (IMAGEOBJPROP IMAGEOBJ 'RULE.HEIGHT)
TOSTREAM])

```

; Edited 5-Jun-91 15:09 by jds
; This function does not build the bitmap but lets the imageboxfn
; cache a bitmap

(CROPMARK.WHENOPERATEDONFN

```

[LAMBDA (A B C C)

])

```

(* gbn " 6-Jan-85 13:23")
(* DUMMY)

(RPAQQ CROPMARK.IMAGE )

(RPAQQ? CROPMARK.DEFAULT.PAGESIZE (LIST 612 792))

(RPAQ CROPMARK.IMAGEFNS

```

(IMAGEFNSCREATE (FUNCTION CROPMARK.DISPLAYFN)
(FUNCTION CROPMARK.IMAGEBOXFN)
(FUNCTION CROPMARK.PUTFN)
(FUNCTION CROPMARK.GETFN)
(FUNCTION CROPMARK.COPYFN)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION NIL)
(FUNCTION CROPMARK.WHENOPERATEDONFN)
(FUNCTION NIL)))

```

FUNCTION INDEX

CROPMARK.COPYFN	6	HRULE.CREATE	1	VRULE.DISPLAYFN	3
CROPMARK.CREATE	5	HRULE.DISPLAYFN	2	VRULE.GETFN	4
CROPMARK.DISPLAYFN	5	HRULE.GETFN	2	VRULE.GETFN2	4
CROPMARK.GETFN	5	HRULE.IMAGEBOXFN	2	VRULE.IMAGEBOXFN	4
CROPMARK.IMAGEBOXFN	5	HRULE.PUTFN	2	VRULE.PUTFN	4
CROPMARK.PUTFN	6	HRULE.WHENOPERATEDONFN	3	VRULE.WHENOPERATEDONFN	4
CROPMARK.WHENOPERATEDONFN	6	VRULE.COPYFN	4		
HRULE.COPYFN	3	VRULE.CREATE	3		

VARIABLE INDEX

CROPMARK.DEFAULT.PAGESIZE	6	HRULE.DEFAULT.WIDTH	3	IMAGEOBJGETFNS	5
CROPMARK.IMAGE	6	HRULE.IMAGEFNS	3	VRULE.DEFAULT.HEIGHT	4
CROPMARK.IMAGEFNS	6	HRULEFNS	3	VRULE.IMAGEFNS	4
