File created: 27-Jan-2024 15:48:33 **{DSK}<home>larry>il>loops>library>GAUGES.;1**

edit by: lmm

changes to: (FNS DrawHTick)

previous date: 15-Aug-90 16:52:11 {DSK}<home>larry>il>loops>library>GAUGES.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ GAUGESCOMS
       ((DECLARE%: DONTCOPY (PROP MAKEFILE-ENVIRONMENT GAUGES)
              (PROP FILETYPE GAUGES))
                                                                ; Copyright (c) Xerox Corporation 1983, 1987
        (CLASSES Gauge GaugeAV)
        (METHODS Gauge.Attach Gauge.AttachProbe Gauge.Attached? Gauge.ChangeFont Gauge.Close Gauge.Destroy
              Gauge.Detach Gauge.DetachProbe Gauge.Reset Gauge.SaveInIT Gauge.SetParameters Gauge.SetScale
              Gauge.Shape Gauge.ShapeToHold Gauge.ShowInstrument Gauge.Update Gauge.Update?)
        (METHODS GaugeAV.DeleteActiveValue GaugeAV.PutWrappedValue GaugeAV.WrappingPrecedence)


;;; Utility functions for Gauges

        (FNS * GAUGEUTILITYFNS)))

(DECLARE%: DONTCOPY

(PUTPROPS GAUGES MAKEFILE-ENVIRONMENT (:PACKAGE "IL" :READTABLE "INTERLISP" :BASE 10))

(PUTPROPS GAUGES FILETYPE :COMPILE-FILE)
)

;; Copyright (c) Xerox Corporation 1983, 1987

(DEFCLASSES Gauge GaugeAV)

(DEFCLASS Gauge (MetaClass AbstractClass doc "An object which presents a dynamic graphical image of a Loops
                     value" Edited%:                            (* edited%: "29-Jan-87 03:28"))
       (Supers Window)
       [ClassVariables (LeftButtonItems ((Update 'Update "Update gauge to show all labels")
                                         ("Save in IT" 'SaveInIT "(SavedValue) and IT _ this gauge")))
              (MiddleButtonItems ((Attached? 'Attached? "Print where gauge attached")
                                  (Detach 'Detach "Detach gauge"]
       (InstanceVariables (reading 0 doc "external value of reading")
              (containedInAV NIL doc "value is active value this gauge is within. That active value should be an
                     instance of the class GaugeAV.")
              (font (MODERN 10)
                    doc "Fonts can be different for each gauge.")
              (width 0 min 0 doc "min property is minimum width a gauge can be. This is computed by
                     SetParameters and is used by the NEWREGION function within a Shape method.")
              (height 0 min 0 doc "min property is minimum height a gauge can be. This is computed by
                     SetParameters and is used by the NEWREGION function within a Shape method.")))

(DEFCLASS GaugeAV (MetaClass Class Edited%:                     (* RBGMartin "19-May-86 16:11")
                        doc "This is the default metaClass for all classes")
       (Supers LocalStateActiveValue)
       (InstanceVariables (gauge NIL doc "the gauge that is connected to this active value")
              (label NIL doc "for barcharts")
              (object NIL doc "The object with the real variable")
              (propName NIL doc "The prop name of the real variable")
              (type NIL doc "The type of the real variable")
              (varName NIL doc "The name of the real variable")))

(\BatchMethodDefs)

(METH Gauge Attach (obj varName propName type xOrPos y)
      "Attach a self to obj/varName"
      (category (Public)))

(METH Gauge AttachProbe (Obj VarName Label PropName Type)
      "Attach a probe to an object, and record attachment here in Gauge. Returns current value of Obj VarName"
      (category (Internal)))

(METH Gauge Attached? (dontPrintFlg)
      "print what gauge is attached to"
      (category (Gauge)))

(METH Gauge ChangeFont (newFont)
      "Change the font of a gauge to the newFont. Update the gauge. Return the old font."
      (category (Gauge)))

(METH Gauge Close (dontDetachFlg)
      "Detach the gauge and close the window"
```

```
      (category (Window)))

(METH Gauge Destroy NIL "Destroy the gauge, detaching it first" (category (Object)))

(METH Gauge Detach NIL "Detach gauge from attachedPlaces it is attached to" (category (Gauge)))

(METH Gauge DetachProbe (av)
      "Delete active value that connects object being monitored to gauge."
      (category (Internal)))

(METH Gauge Reset (newReading)
      "set reading to value, and then update gauge so that it shows that reading without going through
      intermediate states"
      (category (Gauge)))

(METH Gauge SaveInIT NIL "New method template" (category (Gauge)))

(METH Gauge SetParameters NIL "Set the font for the window" (category (Internal)))

(METH Gauge SetScale (min max labelScale)
      "compute the scale and redisplay if necessary"
      (category (Gauge)))

(METH Gauge Shape (newRegion noUpdateFlg)
      "Shapes outside of region to specified shape. Uses GETREGION and width:,min height:,min"
      (category (Public)))

(METH Gauge ShapeToHold NIL "shapes a gauge to its smallest size." (category (Gauge)))

(METH Gauge ShowInstrument NIL "Cause an error, becuase this should be implemented in subclass" (category (
                                                                                                    Internal
                                                                                                      )))

(METH Gauge Update NIL "Set up display window for instrument. Reinitialize to reflect current state"
      (category (Window)))

(METH Gauge Update? NIL "Update if there is a window and it is open" (category (Gauge)))

(Method ((Gauge Attach)
         self obj varName propName type xOrPos y)                ; edited: 27-Jan-87 10:25
       "Attach a self to obj/varName"
       (AND (_ self Attached? T)
            (_ self Detach))
       [OR (@ title)
           (_@
            title
            (COND
               (propName (PACK* varName ":," propName))
               (T varName]
       (_ self Update)
       (OR (CDR (@ containedInAV))
           (_ self Move xOrPos y))
       (_ self AttachProbe obj varName NIL propName type)        ; Move if not attached before
       (_ self Set (GetIt obj varName propName type))
       self)

(Method ((Gauge AttachProbe)
         self Obj VarName Label PropName Type)                   ; edited: 27-Jan-87 22:16
       "Attach a probe to an object, and record attachment here in Gauge. Returns current value of Obj VarName"
       (PROG (gaugeAV)
             (COND
                ((NOT (type? instance Obj))
                 (ERROR Obj " not an instance -- cannot attach a gauge")))
             (SETQ gaugeAV (_ ($ GaugeAV)
                              New))
             (_@
              gaugeAV object Obj)
             (_@
              gaugeAV varName VarName)
             (_@
              gaugeAV label Label)
             (_@
              gaugeAV propName PropName)
             (_@
              gaugeAV type Type)
             (_@
              gaugeAV gauge self)
             (_@
              containedInAV
              (CONS gaugeAV (@ containedInAV)))                  (* CONSing to (@ containedInAV) to handle barcharts)
             (_ gaugeAV AddActiveValue Obj VarName PropName Type)))

[Method ((Gauge Attached?)
         self dontPrintFlg)                                      ; edited: 29-Jan-87 10:29
       "print what gauge is attached to"
       (LET (promptW (containedInAV (@ containedInAV)))
            (COND
```

```
               (dontPrintFlg containedInAV)
               [containedInAV (bind [pW _ (GETPROMPTWINDOW (@ window)
                                                 (ADD1 (TIMES 2 (LENGTH containedInAV)))
                                                 '(MODERN 10]
                                    for AV in containedInAV as rest on containedInAV
                                    first (CLEARW pW)
                                          (printout pW "Attached to:" T)
                                    do (RESETLST
                                           (RESETSAVE (DSPRIGHTMARGIN 1000 pW)
                                                  (LIST 'DSPRIGHTMARGIN (DSPRIGHTMARGIN NIL pW)
                                                        pW))
                                           (printout pW "Obj: " (COND
                                                                    ((GetObjectName (@ AV object)))
                                                                    (T "No Name"))
                                                  T "Var: " (@ AV varName))
                                           [PROG ((propN (@ AV propName)))
                                                 (COND
                                                    (propN (printout pW ":," propN]
                                           (AND (CDR rest)
                                                (printout pW T)))]
               (T [SETQ promptW (GETPROMPTWINDOW (@ window)
                                                 2
                                                 '(MODERN 10]
                  (CLEARW promptW)
                  (printout promptW "Not attached"]

(Method ((Gauge ChangeFont)
         self newFont)                                              ; RBGMartin 28-Apr-86 15:28
       "Change the font of a gauge to the newFont. Update the gauge. Return the old font."
       (PROG1 (@ font)
           (_@
            font newFont)
           (_ self Update)))

(Method ((Gauge Close)
         self dontDetachFlg)                                        ; dgb:  6-Mar-84 15:32
       "Detach the gauge and close the window"
       (AND (NULL dontDetachFlg)
            (_ self Attached? T)
            (_ self Detach))
       (_Super
        self Close))

(Method ((Gauge Destroy)
         self)                                                      ; smL 27-Feb-85 15:54
       "Destroy the gauge, detaching it first"
       (_ self Detach)
       (_Super))

[Method ((Gauge Detach)
         self)                                                      ; edited: 27-Jan-87 22:59
       "Detach gauge from attachedPlaces it is attached to"
       (_ self ClearPromptWindow)
       (COND
          ((@ containedInAV)
           (_ self PromptPrint "Detaching
              ")
           (DISMISS 300)
           (for AV in (@ containedInAV) finally (_@
                                                containedInAV NIL)
              do (_ self DetachProbe AV)))
          (T (_ self PromptPrint "Not Attached
                "]

[Method ((Gauge DetachProbe)
         self av)                                                   ; edited: 28-Oct-86 10:18
       "Delete active value that connects object being monitored to gauge."

;;; The test for DestroyedObject is in here because of a bug in ActiveValue.DeleteNestedValue that will stack overflow if the containingObj argument is a
;;; DestroyedObject

       (COND
          ((OR (_ (@ av object)
                  InstOf
                  ($ DestroyedObject))
               (_ av DeleteActiveValue (@ av object)
                  (@ av varName)
                  (@ av propName)
                  (@ av type)))                                     (* delete any links the gaugeAV may have elsewhere)
           (for iv in (_ av ListAttribute 'IVs) do (PutValueOnly av iv NIL]

(Method ((Gauge Reset)
         self newReading)                                           ; dgb:  6-JUN-83 17:43
       "set reading to value, and then update gauge so that it shows that reading without going through
       intermediate states"
       (_@
        reading newReading)
```

```
       (_ self Update))

(Method ((Gauge SaveInIT)
         self)                                                    ; edited: 27-Jan-87 22:27
       "New method template"
       (PutSavedValue self))

(Method ((Gauge SetParameters)
         self)                                                    ; RBGMartin 29-Apr-86 16:11
       "Set the font for the window"
       (DSPFONT (@ font)
              (@ window)))

(Method ((Gauge SetScale)
         self min max labelScale)                                 ; dgb: 23-JUN-83 15:50
       "compute the scale and redisplay if necessary"
       (_ self ComputeScale min max labelScale)
       (_ self Update?))

(Method ((Gauge Shape)
         self newRegion noUpdateFlg)                              ; RBGMartin  5-May-86 16:11
       "Shapes outside of region to specified shape. Uses GETREGION and width:,min height:,min"
       (PROG ((window (@ window)))
             (CLEARW window)
             (_Super
              self Shape [if newRegion
                             then (GetMinRegion newRegion self)
                           else (GETREGION (@ width%:,min)
                                           (@ height%:,min)
                                           (WINDOWPROP (@ window)
                                                  'REGION]
              noUpdateFlg)))

[Method ((Gauge ShapeToHold)
         self)                                                    ; RBGMartin 30-Apr-86 16:19
       "shapes a gauge to its smallest size."
       ;; JRB - Don't bother if it's not going to change
       (if [OR (NOT (EQUAL (@ width)
                           (@ width%:,min)))
               (NOT (EQUAL (@ height)
                           (@ height%:,min]
           then (_ self Shape (CREATEREGION (@ left)
                                            (@ bottom)
                                            (@ width%:,min)
                                            (@ height%:,min]

(Method ((Gauge ShowInstrument)
         self)                                                    ; edited: 13-NOV-83 18:22
       "Cause an error, becuase this should be implemented in subclass"
       (ERROR "ShowInstrument should be implemented in " (Class self)))

(Method ((Gauge Update)
         self)                                                    ; RBGMartin 19-May-86 12:03
       "Set up display window for instrument. Reinitialize to reflect current state"
       (_Super)
       ;; This apparently extra _Super is here to make sure the window is updated before SetParameters is invoked. The functionality in
       ;; SetParameters may be dependent on the window being sized correctly.
       (_ self Clear)
       (_ self SetParameters)
       (_Super)
       (_ self ShowInstrument)
       (_ self ShowReading)
       self)

(Method ((Gauge Update?)
         self)                                                    ; RBGMartin  7-Apr-86 14:51
       "Update if there is a window and it is open"
       (AND [NOT (NotSetValue (GetIVHere self 'window]
            (OPENWP (GetValue self 'window))
            (_ self Update)))

(\UnbatchMethodDefs)

(\BatchMethodDefs)

(METH GaugeAV DeleteActiveValue (containingObj varName propName type)
      "Remove the active value self from the specified variable. - Return the deleted annotatedValue or NIL if
      the active value was not found"
      (category NIL))

(METH GaugeAV PutWrappedValue (containingObj varName newValue propName type)
      "Send the Set message to the IV gauge of the active value."
      (category NIL))

(METH GaugeAV WrappingPrecedence NIL "What is the wrapping precedence for this active value? The larger the
```

```
      number, the more the active value wants to be on the outside. NIL forces innermost nesting, T forces
      outermost nesting." (category NIL))

(Method ((GaugeAV DeleteActiveValue)
         self containingObj varName propName type)              ; RBGMartin 16-May-86 11:51
       "Remove the active value self from the specified variable. - Return the deleted annotatedValue or NIL if
       the active value was not found"
       (LET ((annoValue (_Super
                         self DeleteActiveValue containingObj varName propName type)))
            NIL annoValue))

(Method ((GaugeAV PutWrappedValue)
         self containingObj varName newValue propName type)     ; edited: 28-Oct-86 21:36
       "Send the Set message to the IV gauge of the active value."
       (_ (@ gauge)
          Set newValue (@ label))                               ; (@ label) arg is needed for BarCharts
       (_Super
        self PutWrappedValue containingObj varName newValue propName type))

(Method ((GaugeAV WrappingPrecedence)
         self)                                                  ; edited: 15-May-86 13:54
       "What is the wrapping precedence for this active value? The larger the number, the more the active value
       wants to be on the outside. NIL forces innermost nesting, T forces outermost nesting." 10)

(\UnbatchMethodDefs)
```

;;; Utility functions for Gauges

```
(RPAQQ GAUGEUTILITYFNS
       (CenterPrint ChangeHorizontalSetting ChangeVerticalSetting ComputeLabels ConcatNSpaces
              DigiMeterNEWREGIONfn DrawBox DrawHTick DrawLines DrawRay DrawTick DrawVTick EvenIntervals
              FloorOfLog GetMinRegion GetPathObj InteriorHeight InteriorWidth LineLeft LineLower LineRaise
              LineRight MeterNEWREGIONfn MoveImageRight PinnedSetting PrintLabelScale RotateLine SetUpBarChart
              ShowRayLabel TellInstrument TitleWidth))

(DEFINEQ
```

(CenterPrint

```
  [LAMBDA (string xpos ypos window)                             (* RBGMartin "29-Apr-86 18:44")
                                                                (* Print string centered around xpos at height ypos)
    (PROG ((font (DSPFONT NIL window)))
          (MOVETO (IDIFFERENCE xpos (LRSH (STRINGWIDTH string font)
                                          1))
                  [IDIFFERENCE ypos (IDIFFERENCE (LRSH (FONTHEIGHT font)
                                                       1)
                                                 (FONTPROP font 'DESCENT]
                  window)
          (PRIN1 string window])
```

(ChangeHorizontalSetting

```
  [LAMBDA (yPos oldXPos newXPos window settingWidth shade)      (* edited%: "29-Oct-86 23:31")
                                                                (* Move a bar to left or right as needed)
    (COND
       ((GREATERP oldXPos newXPos)
        (LineLeft oldXPos newXPos yPos (@ window)
               settingWidth))
       ((GREATERP newXPos oldXPos)
        (LineRight oldXPos newXPos yPos (@ window)
               settingWidth shade])
```

(ChangeVerticalSetting

```
  [LAMBDA (xPos oldYPos newYPos window settingWidth shade)      (* edited%: "29-Oct-86 23:32")
    (COND
       ((GREATERP oldYPos newYPos)
        (LineLower xPos oldYPos newYPos (@ window)
               settingWidth))
       ((GREATERP newYPos oldYPos)
        (LineRaise xPos oldYPos newYPos (@ window)
               settingWidth shade])
```

(ComputeLabels

```
  [LAMBDA (goodMinVal goodIncr goodMaxVal labelScale)           (* edited%: "31-Dec-84 10:30")
    (PROG (incr labelList endPtr lab1 lab)
          [SETQ lab (COND
                       ((GREATERP 1 (ABS goodIncr))
                        goodMinVal)
                       (T                                       (* for increments greater than 1, set the base to be a fixed
                                                                number)
                          (FIX goodMinVal]                      (* For ranges greater than 10, set the increment to a fixed
                                                                number)
          (OR (GREATERP 10 (DIFFERENCE goodMaxVal goodMinVal))
              (SETQ goodIncr (FIX goodIncr)))
      LP  (COND
```

```
            ([AND incr (OR (EQP 0 incr)
                           (COND
                              ((MINUSP incr)
                               (LESSP lab goodMaxVal))
                              (T (GREATERP lab goodMaxVal]
             (GO OUT)))
          (SETQ lab1 (QUOTIENT lab labelScale))
          [COND
             [endPtr (FRPLACD endPtr (SETQ endPtr (LIST lab1]
             (T (SETQ labelList (SETQ endPtr (LIST lab1]
          (SETQ lab (PLUS lab (SETQ incr goodIncr)))
          (GO LP)
      OUT (RETURN labelList])
```

(ConcatNSpaces

```
[LAMBDA (N string)                                          (* dgb%: "25-JAN-83 16:23")
```

(* Put N Spaces on the front of a string. Good for making fixed width strings)

```
   (CONCAT (SELECTQ N
                (0 "")
                (1 " ")
                (2 "  ")
                (3 "   ")
                (4 "    ")
                "     ")
           (COND
              ((IGREATERP N 5)
               (ConcatNSpaces (IDIFFERENCE N 5)
                      string))
              (T string])
```

(DigiMeterNEWREGIONfn

```
[LAMBDA (FP MP self)                                        (* RBGMartin "30-Apr-86 10:06")
```

(* When prompting user for a new region for a DigiMeter, this guarantees the height and width of the "ghost region" are the correct dimensions.)

```
   (LET [(SpaceForLCD (@ spaceForLCD))
         (SpaceForLabelScale (@ spaceForLabelScale))
         (titleHeight (COND
                         ((@ title)
                          (FONTHEIGHT (DSPFONT NIL WindowTitleDisplayStream)))
                         (T 0]
        (COND
           [MP (with POSITION MP (PROG [(DX (IDIFFERENCE XCOORD (fetch (POSITION XCOORD) of FP)))
                                        (DY (IDIFFERENCE YCOORD (fetch (POSITION YCOORD) of FP]
                                       [COND
                                          [(IGREATERP (IABS DX)
                                                  (IDIFFERENCE (IABS DY)
                                                         (IPLUS titleHeight SpaceForLCD SpaceForLabelScale)))
```

(* subtract from DY the height of the LCD, the height of the label, and height of the title.
Check to make sure region for meter is square.)

```
                                           (SETQ YCOORD (IPLUS (fetch (POSITION YCOORD) of FP)
                                                               (ITIMES (IPLUS (IABS DX)
                                                                              titleHeight SpaceForLCD
                                                                              SpaceForLabelScale)
                                                                       (COND
                                                                          ((MINUSP DY)
                                                                           -1)
                                                                          (T 1]
                                          (T (SETQ XCOORD (IPLUS (fetch (POSITION XCOORD) of FP)
                                                                 (ITIMES (IDIFFERENCE (IABS DY)
                                                                                 (IPLUS titleHeight SpaceForLCD
                                                                                        SpaceForLabelScale))
                                                                         (COND
                                                                            ((MINUSP DX)
                                                                             -1)
                                                                            (T 1]
                                       (RETURN MP]
           (T FP])
```

(DrawBox

```
[LAMBDA (left bottom width height lineWidth operation window)  (* dgb%: "29-JAN-83 01:01")
   (PROG ((topy (IPLUS bottom height))
          (topx (IPLUS left width)))                        (* left)
         (DRAWLINE left bottom left topy lineWidth operation window)
                                                            (* bottom)
         (DRAWLINE left bottom topx bottom lineWidth operation window)
                                                            (* right)
         (DRAWLINE topx bottom topx topy lineWidth operation window)
                                                            (* top)
```

```
          (DRAWLINE left topy topx topy lineWidth operation window])
```

(DrawHTick

```
  [LAMBDA (self x y incr)                                   ; Edited 26-Jan-2024 21:00 by lmm
                                                            (* edited%: "22-May-86 13:08")
                                                            ; Draw a horizontal line for a tick
    (PROG ((smallTicks (@ ticks%:,smallTicks))
           (window (@ window))
           (tickLength (@ ticks%:,tickLength)))
          (DRAWLINE x y (IPLUS x tickLength)
                 y
                 (@ brushWidth)
                 'PAINT window)
          (AND incr smallTicks (bind ysmall for N to smallTicks do [SETQ ysmall (IPLUS y
                                                                                       (TIMES N
                                                                                              (FQUOTIENT
                                                                                               incr
                                                                                               (ADD1 smallTicks]
                                                                    (DRAWLINE x ysmall (IPLUS x (IQUOTIENT
                                                                                                 tickLength 2
                                                                                                 ))
                                                                           ysmall
                                                                           (IMAX (QUOTIENT (@ brushWidth)
                                                                                           2)
                                                                                 1)
                                                                           'PAINT window])
```

(DrawLines

```
  [LAMBDA (pointList window fromPointN toPointN width operation color)
                                                            (* dgb%: "25-MAR-83 19:35")
    (OR fromPointN (SETQ fromPointN 1))
    (OR toPointN (SETQ toPointN (LENGTH pointList)))
    (PROG ((pointTail (NTH pointList fromPointN)))
          (OR pointTail (RETURN))
          (MOVETO (fetch XCOORD of (CAR pointTail))
                 (fetch YCOORD of (CAR pointTail))
                 window)
          (for i from (ADD1 fromPointN) by 1 to toPointN as point in (CDR pointTail)
             do (BLOCK)
                (DRAWTO (fetch XCOORD of point)
                       (fetch YCOORD of point)
                       width operation window color])
```

(DrawRay

```
  [LAMBDA (x y length angle width operation window color invisLength)
                                                            (* dgb%: "11-JAN-83 13:12")
          (* Draw a ray from center xy y at angle of length. If invisLength is given, only draw the end of the ray that is visible)

    (PROG ((s (SIN angle))
           (c (COS angle)))
          (DRAWLINE (COND
                       (invisLength (PLUS x (TIMES invisLength c)))
                       (T x))
                 (COND
                    (invisLength (PLUS y (TIMES invisLength s)))
                    (T y))
                 (PLUS x (TIMES length c))
                 (PLUS y (TIMES length s))
                 width operation window color])
```

(DrawTick

```
  [LAMBDA (self angle numExtraTicks incr smallSizeFactor)   (* edited%: "22-May-86 13:13")
                                                            (* Draw a tick and potentially some small ticks)
    (PROG ((tickLength (@ ticks%:,tickLength))
           invisLength)
          (DrawRay (@ xc)
                 (@ yc)
                 (@ radius)
                 angle
                 (@ brushWidth)
                 'PAINT
                 (@ window)
                 NIL
                 (IDIFFERENCE (@ radius)
                        tickLength))
          (OR numExtraTicks (RETURN))                       (* Go on only if there are extra small ticks)
          [SETQ invisLength (IDIFFERENCE (@ radius)
                                  (COND
                                     (smallSizeFactor (TIMES tickLength smallSizeFactor))
                                     (T tickLength]
          (for i to numExtraTicks bind (nt1 _ (ADD1 numExtraTicks))
             do (DrawRay (@ xc)
```

```
                    (@ yc)
                    (@ radius)
                    (PLUS angle (TIMES i (FQUOTIENT incr nt1)))
                    (IMAX (QUOTIENT (@ brushWidth)
                                    2)
                          1)
                    'PAINT
                    (@ window)
                    NIL invisLength])
```

(DrawVTick

```
[LAMBDA (self x y incr)                                    (* edited%: "22-May-86 13:09")
                                                           (* Draw a horizontal line for a tick)
  (PROG ((smallTicks (@ ticks%:,smallTicks))
         (window (@ window))
         (tickLength (@ ticks%:,tickLength)))
        (DRAWLINE x y x (IPLUS y tickLength)
               (@ brushWidth)
               'PAINT window)
        (AND incr smallTicks (bind xsmall for N to smallTicks do [SETQ xsmall (IPLUS x
                                                                                 (TIMES N
                                                                                        (FQUOTIENT
                                                                                         incr
                                                                                         (ADD1 smallTicks]
                                                                  (DRAWLINE xsmall y xsmall
                                                                         (IPLUS y (IQUOTIENT tickLength 2))
                                                                         (IMAX (IQUOTIENT (@ brushWidth)
                                                                                          2)
                                                                               1)
                                                                         'PAINT window])
```

(EvenIntervals

```
[LAMBDA (lower range numIntervals includeLastFlg)          (* edited%: "22-May-86 00:43")

        (* Produces a list of integers in best approximation of an integer linear scale between ower and lower+range.
        Include lower+range only if includeLastFlg=T.)

  (bind (delta _ (FQUOTIENT range numIntervals)) for i from 0 to numIntervals
     when (OR includeLastFlg (NEQ i numIntervals)) collect (PLUS lower (TIMES i delta])
```

(FloorOfLog

```
[LAMBDA (number)                                           (* RBGMartin "29-Apr-86 16:43")
  (LET [(EXPO (FQUOTIENT (LOG number)
                         (LOG 10]                          (* number = 10^EXPO. The COND computes the floor of EXPO.)
       (COND
          ((MINUSP EXPO)
           (FIX (SUB1 EXPO)))
          (T (FIX EXPO])
```

(GetMinRegion

```
[LAMBDA (region self)                                      (* RBGMartin "29-Apr-86 14:15")
                                                           (* returns a region that is at least the minimum width and height.)
  (create REGION using region WIDTH _ (MAX (@ width%:,min)
                                           (fetch WIDTH of region))
                      HEIGHT _ (MAX (@ height%:,min)
                                    (fetch HEIGHT of region])
```

(GetPathObj

```
[LAMBDA (obj path)                                         (* sm%: "12-MAY-83 15:37")
  (COND
     ((NULL path)
      obj)
     (T (GetPathObj (GetValue obj (CAR path))
                    (CDR path])
```

(InteriorHeight

```
[LAMBDA (self)                                             (* edited%: "15-May-86 02:02")
  (IDIFFERENCE (@ height)
         (PLUS (FONTHEIGHT (DSPFONT NIL WindowTitleDisplayStream))
               (TIMES 2 (WINDOWPROP (@ window)
                                    'BORDER])
```

(InteriorWidth

```
[LAMBDA (self)                                             (* RBGMartin "29-Apr-86 15:01")
  (DIFFERENCE (@ width)
         (TIMES 2 (WINDOWPROP (@ window)
                              'BORDER])
```

(LineLeft

```
  [LAMBDA (XA XB Y window height)                                    (* edited%: "15-May-86 00:52")
                                                                     (* Draw a thick line, reducing the size from XB to XA.
                                                                     XA>XB to work)
    (for I from 1 to (IDIFFERENCE XA XB) do (BITBLT NIL NIL NIL window (IDIFFERENCE XA I)
                                                    Y 1 (OR height 4)
                                                    'TEXTURE
                                                    'REPLACE WHITESHADE])
```

(LineLower

```
  [LAMBDA (X YA YB window width)                                     (* edited%: "15-May-86 00:54")
    (for I from 1 to (IABS (IDIFFERENCE YB YA)) do (BITBLT NIL NIL NIL window X (IDIFFERENCE (IMAX YA YB)
                                                                                            I)
                                                           (OR width 4)
                                                           1
                                                           'TEXTURE
                                                           'REPLACE WHITESHADE])
```

(LineRaise

```
  [LAMBDA (X YA YB window width shade)                               (* edited%: "15-May-86 00:54")
    (for I from 0 to (SUB1 (IABS (IDIFFERENCE YB YA))) do (BITBLT NIL NIL NIL window X (IPLUS I (IMIN YA YB))
                                                                  (OR width 4)
                                                                  1
                                                                  'TEXTURE
                                                                  'REPLACE
                                                                  (OR shade BLACKSHADE])
```

(LineRight

```
  [LAMBDA (XA XB Y window height shade)                              (* edited%: "15-May-86 00:52")
    (for I from 0 to (SUB1 (IDIFFERENCE XB XA)) do (BITBLT NIL NIL NIL window (IPLUS I XA)
                                                           Y 1 (OR height 4)
                                                           'TEXTURE
                                                           'REPLACE
                                                           (OR shade BLACKSHADE])
```

(MeterNEWREGIONfn

```
  [LAMBDA (FP MP self)                                               (* RBGMartin "30-Apr-86 16:08")
```

(* When prompting user for a new region for a Meter this guarantees the height and width of the "ghost region" are the correct dimensions.)

```
    (LET ((titleHeight (COND
                         ((@ title)
                          (FONTHEIGHT (DSPFONT NIL WindowTitleDisplayStream)))
                         (T 0)))
          (SpaceForLabelScale (@ spaceForLabelScale)))
         (COND
            [MP (with POSITION MP (PROG [(DX (IDIFFERENCE XCOORD (fetch (POSITION XCOORD) of FP)))
                                         (DY (IDIFFERENCE YCOORD (fetch (POSITION YCOORD) of FP]
                                        [COND
                                           [(IGREATERP (IABS DX)
                                                       (IDIFFERENCE (IABS DY)
                                                              (IPLUS titleHeight SpaceForLabelScale)))
```

(* subtract from DY the height of the label and the title. Check to make sure region for meter is square.)

```
                                            (SETQ YCOORD (IPLUS (fetch (POSITION YCOORD) of FP)
                                                                (ITIMES (IPLUS (IABS DX)
                                                                               titleHeight SpaceForLabelScale)
                                                                       (COND
                                                                          ((MINUSP DY)
                                                                           -1)
                                                                          (T 1]
                                           (T (SETQ XCOORD (IPLUS (fetch (POSITION XCOORD) of FP)
                                                                  (ITIMES (IDIFFERENCE (IABS DY)
                                                                                 (IPLUS titleHeight
                                                                                        SpaceForLabelScale))
                                                                          (COND
                                                                             ((MINUSP DX)
                                                                              -1)
                                                                             (T 1]
                                        (RETURN MP]
              (T FP])
```

(MoveImageRight

```
  [LAMBDA (WINDOW XDELTA)                                            (* dgb%: "11-JAN-83 10:54")
```

(* scrolling function that scrolls by blting existing bits and then filling with background the newly exposed bits. Copied from SCROLLBYREPAINTFN)

```
    (PROG ((DSP (WINDOWPROP WINDOW 'DSP))
           (EXTENT (WINDOWPROP WINDOW 'EXTENT))
           CLIPREG)
```

```
          (SETQ CLIPREG (DSPCLIPPINGREGION NIL DSP))
          (COND
             ((AND (NEQ XDELTA 0)
                   (COND
                      ((AND EXTENT (NEQ (fetch (REGION WIDTH) of EXTENT)
                                        -1))                    (* limit amount by the extent)
                                                                (* for now limit right extent to right of window to keep it always
                                                                visible.)
                       (SETQ XDELTA (IMIN (IDIFFERENCE (fetch (REGION LEFT) of CLIPREG)
                                                       (fetch (REGION LEFT) of EXTENT))
                                          (IMAX (IDIFFERENCE (fetch (REGION PRIGHT) of CLIPREG)
                                                             (fetch (REGION PRIGHT) of EXTENT))
                                                XDELTA)))       (* make sure it is still not 0)
                       (NEQ XDELTA 0))
                      (T T)))
              (BITBLT WINDOW (WTODSX 0 WINDOW)
                     (WTODSY 0 WINDOW)
                     WINDOW
                     (WTODSX XDELTA WINDOW)
                     (WTODSY 0 WINDOW)
                     (fetch (REGION WIDTH) of CLIPREG)
                     (fetch (REGION HEIGHT) of CLIPREG))
              (WXOFFSET XDELTA DSP)
              (FILLWITHBACKGROUND WINDOW (COND
                                           ((IGREATERP XDELTA 0)
                                                                (* moving to right, create new region on left for repaintfn)
                                            (create REGION
                                                    LEFT _ (WTODSX 0 WINDOW)
                                                    BOTTOM _ (WTODSY 0 WINDOW)
                                                    WIDTH _ (IMIN XDELTA (fetch (REGION WIDTH) of CLIPREG))
                                                    HEIGHT _ (fetch (REGION HEIGHT) of CLIPREG)))
                                           (T                   (* moving to left.)
                                              (create REGION
                                                      LEFT _ (IMAX (WTODSX (IPLUS (fetch (REGION WIDTH)
                                                                                     of CLIPREG)
                                                                                  XDELTA)
                                                                           WINDOW)
                                                                   (fetch (REGION LEFT) of CLIPREG))
                                                      BOTTOM _ (WTODSY 0 WINDOW)
                                                      WIDTH _ (IMIN (IMINUS XDELTA)
                                                                    (fetch (REGION WIDTH) of CLIPREG))
                                                      HEIGHT _ (fetch (REGION HEIGHT) of CLIPREG])
```

(PinnedSetting

```
  [LAMBDA (reading minSetting maxSetting)                       (* dgb%: "25-JAN-83 16:31")

          (* Returns the value of reading if it is between minSetting and maxSetting, else the "pinned" value)

    (COND
       ((GREATERP minSetting reading)
        minSetting)
       ((GREATERP reading maxSetting)
        maxSetting)
       (T reading])
```

(PrintLabelScale

```
  [LAMBDA (self)                                                (* dgb%: "28-JAN-83 18:38")
    (COND
       ((NEQ 1 (@ labelScale))
        (MOVETO 1 1 (@ window))
        (PRIN1 'X (@ window))
        (PRIN1 (@ labelScale)
               (@ window])
```

(RotateLine

```
  [LAMBDA (xorg yorg length angle newAngle window width deltaAngle color)
                                                                (* dgb%: "11-JUL-83 14:16")
    (PROG (incr (diff (IDIFFERENCE newAngle angle)))
          (COND
             ((EQUAL angle newAngle)
              (RETURN)))                                        (* First erase the current needle)
          (DrawRay xorg yorg length angle width 'INVERT window color)
          [SETQ incr (COND
                       ((IGREATERP -5 diff)
                        (IMINUS (OR deltaAngle 5)))
                       [(IGREATERP diff 5)
                        (OR deltaAngle (IMAX 5 (IQUOTIENT diff 8]
                       (T (GO end]
          (for a from (IPLUS angle incr) to (IDIFFERENCE newAngle incr) by incr
             do (DrawRay xorg yorg length a width 'INVERT window color)
                                                                (* Draw and then erase a line)
                (DrawRay xorg yorg length a width 'INVERT window color))
      end                                                       (* Now draw final needle)
          (DrawRay xorg yorg length newAngle width 'INVERT window color])
```

(SetUpBarChart

```
  [LAMBDA (listOfObjects ivName chartTitle maxScale path)        (* sm%: "12-MAY-83 15:32")

          (* Attach a bar chart to the iv of specified by the object in the list, and the path.
          path can be an atom, meaning that IV, or it can be a list of atoms to be followed, each an iv name.
          Each bar on the chart will be labelled with the name of the object on listOfObjects.
          title will be used for the BarChart.)

    (bind gaugedObj bc first (_@
                             (SETQ bc (_ ($ BarChart)
                                         New))
                             title chartTitle)
      for obj in listOfObjects do (SETQ gaugedObj (GetPathObj obj path))
                                  (_ bc Attach gaugedObj ivName 'Set (OR (GetObjectName obj)
                                                                         obj))
                                  (AND maxScale (_ bc SetScale 0 maxScale))
      finally (RETURN bc])
```

(ShowRayLabel

```
  [LAMBDA (self angle string)                                    (* edited%: "28-Jan-87 00:44")
    (PROG ((s (SIN angle))
           (c (COS angle)))
          (CenterPrint string (PLUS (@ xc)
                                    [TIMES c (PLUS (@ radius)
                                                   (MAXSTRINGWIDTH (@ labels)
                                                          (@ font]
                                    (COND
                                       ((GREATERP c 0)
                                        2)
                                       (T 0)))
                  [PLUS (@ yc)
                        (TIMES s (PLUS (@ radius)
                                       (QUOTIENT (FONTHEIGHT (@ font))
                                              1.5]
                  (@ window])
```

(TellInstrument

```
  [LAMBDA (self varName newValue propName activeVal type)        (* mjs%: "17-FEB-83 15:11")
                                                                 (* This is a putFn for changing attached instruments)
    (PROG ((temp (PutLocalState activeVal newValue self varName propName type)))
          (for instrument in (GetIt self varName 'myGauges type) do (_ instrument Set newValue))
          (RETURN temp])
```

(TitleWidth

```
  [LAMBDA (str)                                                  (* dgb%: "21-JAN-83 14:44")
    (STRINGWIDTH str (DSPFONT NIL WindowTitleDisplayStream])

)
```

FUNCTION INDEX

VARIABLE INDEX

PROPERTY INDEX