Subject: Re: Widget Hierarchy (fwd)
Posted by Evilio del Rio on Thu, 15 Jan 1998 08:00:00 GMT
View Forum Message <> Reply to Message

On Thu, 15 Jan 1998, Chris and Cathy Campo wrote:
> Please repost your IDL routines in ASCII text format. Your attached
> routines came out as "gibberish" on my newsreader.
>
> Thank you, Cathy
>

Sorry Cathy. My mail agent has translated the attachment in MIME-format. It's quite standard and anyone with a modern MIME-compliant news/mail reader will have no problem.

I thought that it would be better to repost them only to you. The repost got back to me because it seems that you are connected in a strange way and you don't have a good return e-mail address (not even a real domain name either). Anyway, here they are.

I mark each file with a header with the name in it, you must to separete them adecuately by hand.

Regards,
Evilio

```
; cut here
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; ;;;;;;;;;;;
;
; viewplot__define.pro
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; ;;;;;;;;;;;
;$Id: viewplot__define.pro,v 1.1 1998/01/14 10:41:49 edelrio Exp edelrio $
;
; Init
;
function ViewPlot::Init

   self.oView  = OBJ_NEW('IDLgrView')
   self.oModel = OBJ_NEW('IDLgrModel')
   self.oPlot  = OBJ_NEW('IDLgrPlot')
   self.oAxX   = OBJ_NEW('IDLgrAxis')
   self.oAxY   = OBJ_NEW('IDLgrAxis')
   self.oTitle = OBJ_NEW('IDLgrText','')
   self.oMisc  = OBJ_NEW('IDL_Container')

   self.oView->Add,self.oModel
   self.oModel->Add,self.oPlot
```

```
  self.oModel->Add,self.oAxX
  self.oModel->Add,self.oAxY
  self.oModel->Add,self.oTitle

  white = [255B,255B,255B]
  black = [  0B,  0B,  0B]
  tf    = '(G8.2)'


  self.oView->SetProperty,COLOR=black

  self.oPlot->SetProperty,/HISTOGRAM,COLOR=white

  self.oAxX->SetProperty,DIRECTION=0,/EXACT,COLOR=white,$
   MAJOR=3,MINOR=0,TICKF=tf

  self.oAxY->SetProperty,DIRECTION=1,/EXACT,COLOR=white,$
   MAJOR=3,MINOR=0,TICKF=tf

  self.oTitle->SetProperty,ALIGN=0.5,COLOR=white,/ONGLASS

  return,1
end



;
; Clean Up
;
pro ViewPlot::CleanUp
  OBJ_DESTROY,[self.oView,self.oModel,self.oPlot, $
            self.oAxX, self.oAxY,  self.oTitle,self.oMisc]
  return
end



;
; SetProperty
;
pro  ViewPlot::SetProperty,DataX=dataX,DataY=dataY,Xrange=xr,Yran ge=yr,$
          Linestyle=listy,Title=tit,Xtitle=xtit,Ytitle=ytit,HISTOGRAM= his

  DoSetView = 0B

  if (N_Elements(datay) GE 1L) then begin
    self.oPlot->SetProperty,DATAY=datay
    DoSetView = 1B
  endif
```

```
  if (N_Elements(datax) GE 1L) then begin
    self.oPlot->SetProperty,DATAX=datax
    DoSetView = 1B
  endif

  if (N_Elements(xr) EQ 2L) then begin
    self.oPlot->SetProperty,XRANGE=xr
    self.oAxX->SetProperty,RANGE=xr
    DoSetView = 1B
  endif

  if (N_Elements(yr) EQ 2L) then begin
    self.oPlot->SetProperty,YRANGE=yr
    self.oAxY->SetProperty,RANGE=yr
    DoSetView = 1B
  endif

  if (N_Elements(his) EQ 1L) then self.oPlot->SetProperty,HISTOGRAM=his

  if (N_Elements(listy) EQ 1L) then self.oPlot->SetProperty,LINESTYLE=listy

  if (N_Elements(tit)   GE 1L) then self.oTitle->SetProperty,STRING=tit

  if (N_Elements(xtit)  GE 1L) then begin
    self.oAxX->GetProperty,TITLE=oXtit
    if (Not(OBJ_VALID(oXtit))) then begin
      oXtit = OBJ_NEW('IDLgrText')
      self.oAxX->SetProperty,TITLE=oXtit
      self.oMisc->Add,oXtit
    endif
    oXtit->SetProperty,STRING=xtit
  endif

  if (N_Elements(ytit)  GE 1L) then begin
    self.oAxY->GetProperty,TITLE=oYtit
    if (Not(OBJ_VALID(oYtit))) then begin
      oYtit = OBJ_NEW('IDLgrText')
      self.oAxY->SetProperty,TITLE=oYtit
      self.oMisc->Add,oYtit
    endif
    oYtit->SetProperty,STRING=ytit
  endif

  if (DoSetView) then self->SetView

  return
end
```

```
;
; GetProperty
;
pro  ViewPlot::GetProperty,DataX=dataX,DataY=dataY,Xrange=xr,Yran ge=yr,$
           Linestyle=listy,Title=tit,Xtitle=xtit,Ytitle=ytit,HISTOGRAM= his

  self.oPlot-> GetProperty,DATA=data,XRANGE=xr,YRANGE=yr,LINESTYLE=listy,HI
STOGRAM=his
  datax = data[0L,*]
  datay = TEMPORARY(data[1L,*])

  self.oTitle->GetProperty,STRING=tit

  self.oAxX->GetProperty,TITLE=oXtit
  oXtit->GetProperty,STRING=xtit

  self.oAxY->GetProperty,TITLE=oYtit
  oYtit->GetProperty,STRING=ytit

  return
end




;
; GetComponent
;
pro  ViewPlot::GetComponent,VIEW=v,MODEL=m,PLOT=p,AXX=x,AXY=y,TIT LE=t,MISC=o
  v = self.oView
  m = self.oModel
  p = self.oPlot
  x = self.oAxX
  y = self.oAxY
  t = self.oTitle
  o = self.oMisc
  return
end


;
; SetView
;
pro ViewPlot::SetView
  self.oPlot->GetProperty,XRANGE=xrange,YRANGE=yrange
  dx = xrange[1]-xrange[0]
  dy = yrange[1]-yrange[0]
```

```
   if (dx LE 0.0) then xrange[1]=xrange[1]+0.1
   if (dy LE 0.0) then yrange[1]=yrange[1]+0.1
   self.oAxX-> SetProperty,RANGE=xrange,LOCATION=[xrange[0],yrange[0]],TICK LEN=dy*0.02
   self.oAxY-> SetProperty,RANGE=yrange,LOCATION=[xrange[0],yrange[0]],TICK LEN=dx*0.02

   self.oTitle->SetProperty,$
    LOCATION=[dx/2.0,1.1*dy]+[xrange[0],yrange[0]],$
    CHAR_DIM= 0.05*[dx,dy]

   self.oAxX->GetProperty,TICKTEXT=oTxtAx,TITLE=oXtit
   oTxtAx->SetProperty,CHAR_DIM= 0.05*[dx,dy]
   if (OBJ_VALID(oXtit)) then oXtit->SetProperty,CHAR_DIM= 0.05*[dx,dy]

   self.oAxY->GetProperty,TICKTEXT=oTxtAy,TITLE=oYtit
   oTxtAy->SetProperty,CHAR_DIM= 0.05*[dx,dy]
   if (OBJ_VALID(oYtit)) then oYtit->SetProperty,CHAR_DIM= 0.05*[dy,dx]

   vpr = Transpose([[xrange],[yrange]])
   vpr[*,1]=vpr[*,1]-vpr[*,0]
   margin = vpr & margin[*,0] = -0.5*vpr[*,1]
   margin[0,*]=  0.5*margin[0,*]
   margin[1,*]=  0.5*margin[1,*]
   vpr = vpr+margin
   self.oView->SetProperty,ViewPlane=vpr[*]

   return
end



;
; SonMethods
;
FUNCTION ViewPlot::SonMethod,son,method,ASFUNCTION=ret
   on_error,2
   cmd =  'self.'+STRTRIM(son,2)+'->'+STRTRIM(method,2)
   IF (ARG_PRESENT(ret)) THEN cmd =  'ret = '+cmd
   print,cmd
   ok =  EXECUTE(cmd)
   return,ok
end



;
; Draw
;
```

```
pro ViewPlot::Draw,oWindow
 ok = OBJ_VALID(oWindow)
;
;              W A R N I N G
; Superclass IDLGRSRCDEST is NOT documented by IDL and
; it may change in future versions.
;
 if (ok) then ok = ok*OBJ_ISA(oWindow,'IDLGRSRCDEST')
 if (ok) then begin
   if (OBJ_ISA(oWindow,'IDLGRPRINTER')) then begin
; Make a B&W plot to the printer
    printbw,self.oView,oWindow
  endif else begin
   oWindow->Draw,self.oView
 endelse

 endif

return
end









;
; Define
;
pro viewplot__define
   viewplot = {VIEWPLOT, $
             oView  : OBJ_NEW(), $
             oModel : OBJ_NEW(), $
             oPlot  : OBJ_NEW(), $
             oAxX   : OBJ_NEW(), $
             oAxY   : OBJ_NEW(), $
             oTitle : OBJ_NEW(), $
             oMisc  : OBJ_NEW()  $
          }
end
;;; End of viewplot__define.pro (cut here)

 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; ;;;;;;;;;;;
;
; viewplot.pro
```

```
;
 ..................................................................... ............
 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; ;;;;;;;;;;;
;$Id: viewplot.pro,v 1.3 1998/01/14 10:43:21 edelrio Exp edelrio $

;
;
;
;
pro vpoptions,oVP
  TheClass = 'VIEWPLOT'
  ok = OBJ_VALID(oVP)
  if (ok) then ok = ok*OBJ_ISA(oVP, TheClass)
  if (Not(ok)) then message,'Not a valid '+TheClass+' object.'
  oVP->GetProperty,Xrange=xr,Yrange=yr, $
   Linestyle=listy,Title=tit,Xtitle=xtit, $
   Ytitle=ytit,HISTOGRAM=his


  MenuDesc = [ $
            '1, BASE, , COLUMN, /FRAME',$
            '0, LABEL, X Range, CENTER', $
            '1, BASE , ,ROW   , FRAME' , $
            '0, FLOAT, '+STRN(xr[0L])+', LABEL_TOP=XMIN, TAG=XMIN', $
            '2, FLOAT, '+STRN(xr[1L])+', LABEL_TOP=XMAX, TAG=XMAX', $
            '0, LABEL, Y Range, CENTER', $
            '1, BASE , ,ROW   , FRAME' , $
            '0, FLOAT, '+STRN(yr[0L])+', LABEL_TOP=YMIN, TAG=YMIN', $
            '2, FLOAT, '+STRN(yr[1L])+', LABEL_TOP=YMAX, TAG=YMAX', $
            '0, LABEL, Titles, CENTER', $
            '1, BASE , ,COLUMN   , FRAME' , $
            '0, TEXT, '+tit+', LABEL_TOP=Plot Title, TAG=tit, WIDTH='+STRN(STRLEN(tit)), $
            '0, TEXT, '+xtit+', LABEL_TOP=X Title, TAG=xtit, WIDTH='+STRN(STRLEN(xtit)), $
            '2, TEXT, '+ytit+', LABEL_TOP=Y Title, TAG=ytit, WIDTH='+STRN(STRLEN(ytit)), $
            '0, INTEGER, '+STRN(his)+', LABEL_TOP=Histogram, TAG=HIST',$
            '2, INTEGER, '+STRN(listy)+', LABEL_TOP=Line Style, TAG=LINE',$
            '0, BUTTON, OK, QUIT, TAG=OK', $
            '2, BUTTON, Cancel, QUIT, TAG=CANCEL' $
           ]

  P = CW_FORM(MenuDesc,TITLE='Plot Options',/COLUMN)

  if (P.ok EQ 1L) then Begin
    xr = [p.xmin,p.xmax]
    yr = [p.ymin,p.ymax]
    listy = p.line
    his   = p.hist
    tit   = p.tit
    xtit  = p.xtit
    ytit  = p.ytit
```

```
      oVP->SetProperty,Xrange=xr,Yrange=yr, $
       Linestyle=listy,Title=tit,Xtitle=xtit, $
       Ytitle=ytit,HISTOGRAM=his
   endif

   return
end



;
;
;
pro viewplot_event,Ev
   WIDGET_CONTROL,Ev.Id,GET_VALUE=value
   WIDGET_CONTROL,Ev.Top,GET_UVALUE=oVP

   case Value of
      'Options' : VPOPTIONS,oVP[0L]
      'Print'   : Begin
                  ; Create Printer
                  oPrint = OBJ_NEW('IDLgrPrinter')
                  ; Ask user for printer.
                  ok =  DIALOG_PRINTERSETUP(oPrint,DIALOG_PARENT=ev.top,TITLE='Print er
Setup')
                  ok =  ok*DIALOG_PRINTJOB(oPrint,DIALOG_PARENT=ev.top,TITLE='Print Job
Setup')
                  ; If ok do the Printing
                  if (ok) then Begin
                     WIDGET_CONTROL,/HOURGLASS
                     oVP[0L]->GetComponent,View=oV
                     PRINTBW,oV,oPrint
                     oPrint->NewDocument
                     OBJ_DESTROY,oPrint
                  endif
                 End

      'Close'   : Begin
                  WIDGET_CONTROL,Ev.Top,/DESTROY
                  return
                 End
   endcase

   WIDGET_CONTROL,/HOURGLASS
   oVP[0L]->Draw,oVP[1L]

   return
end
```

```
;
;
;
pro  viewplot,oViewPlot,TITLE=title,GROUP_LEADER=grpl,XSIZE=xsize ,YSIZE=ysize

   TheClass = 'VIEWPLOT'
   ok = OBJ_VALID(oViewPlot)
   if (ok) then ok = ok*OBJ_ISA(oViewPlot, TheClass)
   if (Not(ok)) then message,'Not a valid '+TheClass+' object.'
   if (Keyword_Set(grpl)) then modal = 1B else modal = 0B

   wbase =  WIDGET_BASE(GROUP_LEADER=grpl,ROW=4,TITLE=title,MODAL=modal)
        wdraw = WIDGET_DRAW(  wbase,XSIZE=xsize,YSIZE=ysize,GRAPHICS_LEVEL=2)
        wopts = WIDGET_BUTTON(wbase,XSIZE=xsize,VALUE='Options')
        woprn = WIDGET_BUTTON(wbase,XSIZE=xsize,VALUE='Print')
        wocls = WIDGET_BUTTON(wbase,XSIZE=xsize,VALUE='Close')

   WIDGET_CONTROL,wbase,/REALIZE
   WIDGET_CONTROL,wdraw,GET_VALUE=oWin
   WIDGET_CONTROL,wbase,SET_UVALUE=[oViewPlot,oWin]

   oViewPlot->Draw,oWin

   XMANAGER,TheClass,wbase

   return
end
;;; End of viewplot.pro
```