
Subject: 02:Re:Table Widget

Posted by [UUCP](#) on Fri, 23 Jan 1998 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

From: DMottershead <DMottershead@mhl.nsw.gov.au>

Subject: Re:Table Widget

Organization: Manly Hydraulics Laboratory

```
(*ptr).plot.xAxis1->SetProperty, tickdir = 1
(*ptr).plot.xAxis1->SetProperty, Location = [0,0]
;(*ptr).plot.xAxis1->SetProperty, Location = [xs[0],ys[0]]
(*ptr).plot.yAxis1->SetProperty, Ticklen = 0.01
;(*ptr).plot.xAxis1->SetProperty, major = ymajor
(*ptr).plot.yAxis1->SetProperty, minor = 4
(*ptr).plot.yAxis1->SetProperty, range = yrange
(*ptr).plot.yAxis1->SetProperty, YCoord_Conv = ys
(*ptr).plot.yAxis1->SetProperty, tickdir = 1
;(*ptr).plot.yAxis1->SetProperty, Location = [xs[0],ys[0]]
(*ptr).plot.yAxis1->SetProperty, Location = [0,0]

;(*ptr).plot.xAxis1->GetProperty, Ticktext = xAxisText
;(*ptr).plot.yAxis1->GetProperty, Ticktext = yAxisText

xaxisText = Obj_New('IDLgrText', font = (*ptr).plot.helvetica6pt)
yaxisText = Obj_New('IDLgrText', font = (*ptr).plot.helvetica6pt)

xAxisText->SetProperty, Font=(*ptr).plot.helvetica6pt
yAxisText->SetProperty, Font=(*ptr).plot.helvetica6pt

(*ptr).obj.odel->add, xaxisText
(*ptr).obj.odel->add, yaxisText

; render the graphics to the window

(*ptr).obj.owindow -> Draw, (*ptr).obj.oview
;endif else begin
;help, result2
;a = result2
;print, a
;return
;endelse

;Put the info structure back.

;Widget_Control, event.top, Set_UValue=ptr, /No_Copy

;case uval of
;'insertrows': begin
```

```

;result = widget_info(event.id,/table_select)
;print, 'ir', result
;result = widget_info(event.id,/table_select)
;widget_control, insert_rows

;return
;end
;insertcols': begin

;result = widget_info(event.id,/table_select)
;print, 'ic', result
;result = widget_info(event.id,/table_select)
;widget_control, insert_rows

return
;end
;endcase

END

pro ctdplt_event, event

WIDGET_CONTROL, event.top, GET_UVALUE = ptr
WIDGET_CONTROL, event.id, GET_UVALUE = uval

case uval of
'exit':begin
    ptr_free, ptr
    WIDGET_CONTROL, event.top, /DESTROY
    return
end

'draw':begin
    WIDGET_CONTROL, event.top, /realize
    return
end

'insertrows': begin

result = CTD_Table_Widget_Events(event)

return
end

'insertcols': begin

```

```

result = widget_info(event.id,/table_select)

return
end

'IDLhelp': begin
    online_help
end

'printsetup': begin
    result = Dialog_PrinterSetup((*ptr).thisPrinter)
end

'print': begin
    dstatus = DIALOG_MESSAGE(/QUESTION, $
        'Printing Can Take a long time on a windows printer.
Continue??')
    if (strucase(dstatus) eq 'YES') then begin
        result = Dialog_PrintJob((*ptr).thisPrinter)
        IF result EQ 1 THEN BEGIN
            (*ptr).thisPrinter->Draw, (*ptr).obj.oview
            (*ptr).thisPrinter->NewDocument
        ENDIF
    endif
    WIDGET_CONTROL, event.top, /realize
end

endcase
return
end

pro ctdplt

data = readgen()

help, data, /structures,output = out

b = n_tags(data)
c = n_elements(data.field01)

; will do for now

plotdata = dblarr(b,c)
i = 0
if i lt b then plotdata(i,*) = data.field01
i = i + 1

```

```

if i lt b then plotdata(i,*) = data.field02
i = i + 1
if i lt b then plotdata(i,*) = data.field03
i = i + 1
if i lt b then plotdata(i,*) = data.field04
i = i + 1
if i lt b then plotdata(i,*) = data.field05
i = i + 1
if i lt b then plotdata(i,*) = data.field06
i = i + 1
if i lt b then plotdata(i,*) = data.field07
i = i + 1
if i lt b then plotdata(i,*) = data.field08
i = i + 1
if i lt b then plotdata(i,*) = data.field09
i = i + 1
if i lt b then plotdata(i,*) = data.field10
i = i + 1
if i lt b then plotdata(i,*) = data.field11
i = i + 1
if i lt b then plotdata(i,*) = data.field12
i = i + 1
if i lt b then plotdata(i,*) = data.field13
i = i + 1
if i lt b then plotdata(i,*) = data.field14
i = i + 1
if i lt b then plotdata(i,*) = data.field15
i = i + 1
if i lt b then plotdata(i,*) = data.field16
i = i + 1

```

```

tlb = Widget_Base(TITLE = 'CTD', $
    /COLUMN, $
    MBAR = menubar)

```

```

fileRow = WIDGET_BASE(tlb, $
    /ROW, $
    SPACE = 20)

```

```

fileTable = WIDGET_TABLE(tlb, $
    /frame, $
    scr_Xsize = 640, $
    scr_ysize = 190, $
    alignment = 0, $
    /scroll, $
    /editable, $
    /resizeable_columns, $
    /resizeable_rows, $

```

```
    /all_events, $  
    VALUE = plotdata, $  
    Event_Pro='CTD_Table_Widget_Events', $  
    UVALUE = 'tablebut')
```

```
; Application Menu Bar
```

```
fileMenu = WIDGET_BUTTON(menubar, $  
    VALUE = 'File', $  
    /MENU)  
printMenBut = WIDGET_BUTTON(fileMenu, $  
    VALUE = 'Print', $  
    UVALUE = 'print')  
printsetupMenBut = WIDGET_BUTTON(fileMenu, $  
    VALUE = 'Print Setup..', $  
    UVALUE = 'printsetup')  
exitMenBut = WIDGET_BUTTON(fileMenu, $  
    VALUE = 'Exit', $  
    UVALUE = 'exit', $  
    /SEPARATOR)
```

```
tableMenu = WIDGET_BUTTON(menubar, $  
    VALUE = 'Table', $  
    /MENU)  
insertrowBut = WIDGET_BUTTON(tableMenu, $  
    VALUE = 'Insert Rows', $  
    UVALUE = 'insertrows')  
insertcolBut = WIDGET_BUTTON(tableMenu, $  
    VALUE = 'Insert Columns', $  
    UVALUE = 'insertcols')
```

```
helpMenu = WIDGET_BUTTON(menubar, $  
    VALUE = 'Help', $  
    /MENU, $  
    /HELP)  
idlhelpBut = WIDGET_BUTTON(helpMenu, $  
    VALUE = 'Help on IDL', $  
    UVALUE = 'IDLhelp')
```

```
WIDGET_CONTROL, tlb, /realize
```

```
; find the screen size
```

```
device, get_screen_size = scr  
if (n_elements(xdim) eq 0) then xdim = fix(scr[0] * 0.8)  
if (n_elements(ydim) eq 0) then ydim = fix(scr[1] * 0.5)
```

```

; create the draw area that will contain the
; object graphics

objectDraw = WIDGET_DRAW(tlb, $
    XSIZE = xdim, $
    YSIZE = ydim, $
    GRAPHICS_LEVEL = 2, $
    UVALUE = 'draw', $
    /EXPOSE_EVENTS)

; realize the control heirarchy

WIDGET_CONTROL, tlb, /REALIZE

; retrieve the object window ID from the drea area ID

WIDGET_CONTROL, objectDraw, GET_VALUE = oWindow

; add the view to the window
; create the printer object

thisPrinter = obj_new('IDLgrPrinter')

; create a view

oview = obj_new('IDLgrView', $
    color = [0,0,0], $
    viewplane_rect = [-0.2,-0.2,1.4,1.4], $
    location = [0,0], $
    units = 3)

; create a model

omodel = obj_new('IDLgrModel')

; add the model to the view

oview -> add, omodel

; set up default symbol

psym = Obj_New('IDLgrSymbol', Color=[255,0,0], Size=0.05)

; default fonts

helvetica12pt = Obj_New('IDLgrFont', 'Helvetica', Size=12)
helvetica10pt = Obj_New('IDLgrFont', 'Helvetica', Size=10)
helvetica8pt = Obj_New('IDLgrFont', 'Helvetica', Size=8)

```

```

helvetica6pt = Obj_New('IDLgrFont', 'Helvetica', Size=6)

; default title strings
; create default plot title

; create x and y axis titles

thisPlot = Obj_New('IDLgrPlot', Color=[0,255,0])

; create the x and y axes

axiscolor = [255,255,0]

xAxis1 = Obj_New('IDLgrAxis', 0, Color=axiscolor, tickdir = 1, ticklen =
0.01, $
    minor = 4, /exact)

yAxis1 = Obj_New('IDLgrAxis', 1, Color=axiscolor, tickdir = 1, ticklen =
0.01, $
    minor = 4, /exact)

oModel->add, yAxis1
oModel->add, xAxis1
oModel->Add, thisPlot

; create the information structure

plot = {thisplot:thisplot, $
    helvetica6pt:helvetica6pt, $
    ;xaxistext:xaxistext, $
    ;yaxistext:yaxistext, $
    xaxis1:xaxis1, $
    yaxis1:yaxis1}

; objects structure

obj = {oWindow:oWindow, $
    oview:oview, $
    omodel:omodel}

info = {obj:obj, $
    data:data, $
    plotdata:plotdata, $
    plot:plot}

; pointer to info structures

ptr = ptr_new(info, /no_copy)

```

```
; put the information structure into the UVALUE of  
; the top-level base
```

```
WIDGET_CONTROL, tlb, SET_UVALUE = ptr
```

```
; call Xmanager to start up the main event loop
```

```
Xmanager, 'ctdplt', tlb
```

```
Return
```

```
end
```

```
--
```

```
Regards
```

```
David
```

```
*****
```

```
David Mottershead      Phone: +61 2 9949 0234  
Manly Hydraulics Laboratory  Fax: +61 2 9948 6185  
110b King St, Manly Vale, 2093 email: dmottershead@mhl.nsw.gov.au  
SYDNEY, AUSTRALIA      WWW: http://www.mhl.nsw.gov.au
```

```
*****
```

```
--
```

```
|Fidonet: UUCP 2:500/3.1
```

```
|Internet: UUCP@p1.f3.n500.z2.hccfido.hcc.nl
```

```
|
```

```
| Standard disclaimer: The views of this user are strictly his own.
```
