
Subject: Re: ZOOM in on DATA
Posted by [davidf](#) on Thu, 14 May 1998 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Lou Youssef (youssef_l@my-dejanews.com) writes:

> Ok, i got my zoom procedure to work but it won't zoom in on data that i plot
> on a map is there a way to do this

Here is a short little example program that shows you how to zoom into a map projection. You drag a rubberband box around the region you want to zoom into. If you want to go back to the full projection, just click anywhere in the window.

Cheers,

David

```
,***** *  
; PRO MapZoom_Widget_Events, event  
  
; This is the event handler for the draw widget graphics window.  
  
; Deal only with DOWN, UP, and MOTION events.  
  
IF event.type GT 2 THEN RETURN  
  
; Get the info structure.  
  
Widget_Control, event.top, Get_UValue=info, /No_Copy  
  
; What kind of event is this?  
  
eventTypes = ['DOWN', 'UP', 'MOTION']  
thisEvent = eventTypes[event.type]  
  
CASE thisEvent OF  
  
  'DOWN': BEGIN  
  
    ; Turn motion events on for the draw widget.  
  
    Widget_Control, info.drawID, Draw_Motion_Events=1  
  
    ; Create a pixmap. Store its ID. Copy window contents into it.  
  
    Window, /Free, /Pixmap, XSize=info.xsize, YSize=info.ysize
```

```

info.pixID = !D.Window
Device, Copy=[0, 0, info.xsize, info.ysize, 0, 0, info.wid]

; Get and store the static corner of the box.

info.sx = event.x
info.sy = event.y

ENDCASE

'UP': BEGIN

; Erase the last box drawn. Destroy the pixmap.

WSet, info.wid
Device, Copy=[0, 0, info.xsize, info.ysize, 0, 0, info.pixID]
WDelete, info.pixID

; Turn draw motion events off.
; Clear any events queued for widget.

Widget_Control, info.drawID, Draw_Motion_Events=0, Clear_Events=1

; Order the box coordinates.

sx = Min([info.sx, event.x], Max=dx)
sy = Min([info.sy, event.y], Max=dy)

; Make sure there was a drag, or just zoom out to
; full size.

IF Abs(dx - sx) LT 10 OR Abs(dy - sy) LT 10 THEN BEGIN

Map_Set, /Cylindrical, /Grid, /Continent, /Isotropic, /Label

ENDIF ELSE BEGIN

; Convert the coordinates to map data coordinates.

!X = info.xscale
!Y = info.yscale
!Map = info.map
latlon = Convert_Coord([sx,dx],[sy,dy], /Device,/To_Data)
loncenter = ((latlon[0,1] - latlon[0,0]) / 2.0) + latlon[0,0]
latcenter = ((latlon[1,1] - latlon[1,0]) / 2.0) + latlon[1,0]

; Draw the map.

```

```
Map_Set, latcenter, loncenter, /Cylindrical, $  
/Grid, /Continent, /Isotropic, /Label, $  
Limit=[latlon[1,0], latlon[0,0], latlon[1,1], latlon[0,1]]
```

```
ENDELSE
```

```
; Update the data scaling parameters.
```

```
info.xscale = !X  
info.yscale = !Y  
info.map = !Map
```

```
ENDCASE
```

```
'MOTION': BEGIN
```

```
; Here is where the actual box is drawn and erased.  
; First, erase the last box.
```

```
WSet, info.wid  
Device, Copy=[0, 0, info.xsize, info.ysize, 0, 0, info.pixID]
```

```
; Get the coordinates of the new box and draw it.
```

```
sx = info.sx  
sy = info.sy  
dx = event.x  
dy = event.y  
PlotS, [sx, sx, dx, dx, sx], [sy, dy, dy, sy, sy], /Device, $  
Color=info.boxColor
```

```
ENDCASE
```

```
ENDCASE
```

```
; Store the info structure.
```

```
Widget_Control, event.top, Set_UValue=info, /No_Copy  
END  
;-----
```

```
PRO MapZoom
```

```
; This is the widget definition module for the program.
```

```
xsize = 500
```

ysize = 300

; Create the TLB.

tlb = Widget_Base(Title='Zooming into Map Example Program')

; Create the draw widget graphics window. Turn button events ON.

drawID = Widget_Draw(tlb, XSize=xsize, YSize=ysize, Button_Events=1)

; Realize widgets and make draw widget the current window.

Widget_Control, tlb, /Realize

Widget_Control, drawID, Get_Value=wid

WSet, wid

; Load drawing color and display the initial map projection.

boxColor = !D.N_Colors-2

TVLCT, 255, 255, 0, boxColor

Map_Set, /Cylindrical, /Grid, /Continents, /Isotropic, /Label

; Create an "info" structure with information to run the program.

```
info = { wid:wid, $          ; The window index number.  
        drawID:drawID, $    ; The draw widget identifier.  
        pixID:-1, $         ; The pixmap identifier (undetermined now).  
        xsize:xsize, $      ; The X size of the graphics window.  
        ysize:ysize, $      ; The Y size of the graphics window.  
        sx:-1, $           ; The X static corner of the box.  
        sy:-1, $           ; The Y static corner of the box.  
        xscale:!X, $        ; The X data scaling parameters.  
        yscale:!Y, $        ; The Y data scaling parameters.  
        map:!Map, $         ; The map scaling parameters.  
        boxColor:boxColor } ; The rubberband box color.
```

; Store the info structure.

Widget_Control, tlb, Set_UValue=info, /No_Copy

; Start the program going.

XManager, 'mapzoom', tlb, /No_Block, \$

Event_Handler='MapZoom_Widget_Events'

END

David Fanning, Ph.D.

Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
