Subject: Success!! Add and manipulate a visualization in standard iTool - Part II
Posted by kidpix on Fri, 20 May 2011 11:28:18 GMT
View Forum Message <> Reply to Message

After a lot of learning-by-doing (where doing means take the manual, search for undocumented class, open the class definition and read it there), I succeed in the post topic: add and manipulate a visualization in standard iTool.

The acutal iTool is an iPlot (IDLitTOOLPLOT) and the visualization is a IDLitVisPolyline encapsulating a IDLgrPolyline objects.

The resulting IDLitVisPolyline obeject in the iTool is working as desired: user can manipulate all the properties via the Visualization Browser and via command line or programmatically using an IDL program or a widget.

I try the same for IDLitVisPolygon object, but the interactive manipulation is somehow forbidden. From the iTool's Visualization Browser I can just change the STYLE property, but thing like COLOR or LINE_STYLE are unchangeable.

Do anyone have experience on this particular point?

That's my example code, where I'm using Coyote CGS functions too (thanks David!) for the polylines RGB color code.

```
;-- IDL code Area

iplot,[-2,3],[-2,4],IDENTIFIER=plotID,/overplot
idTool = ITGETCURRENT(TOOL=oTool) ;- redundant if we are going to work on the current iTool
 oPlot=oTool->IDLitContainer::GetByIdentifier(plotID)

;- 2 ROIs generation
X_roi=[0,0,1,1,0]
Y_roi=[0,1,1,0,0]
X_roi=[X_roi,X_roi+2]
Y_roi=[Y_roi,Y_roi+2]

;- Definition of ROI's number and shape
n_poly= 2
n_vertex=5

;- Generation of connectivity Array based on input parameter
poly_unit=indgen(n_vertex)
poly_vertex_pos=indgen(n_poly)*n_vertex
POLYLINES=[n_vertex,poly_unit+poly_vertex_pos(0)]
  for pl=1,n_poly-1 do  POLYLINES=[POLYLINES,n_vertex,poly_unit+poly_vertex_pos(pl)]

;- Instantiation of polyline (IDLgrPolyline) object, setting of calculated POLYLINES property
  oPolyl =  OBJ_NEW('IDLgrPolyline',X_roi,Y_roi,POLYLINES=POLYLINES,/REG
```

```
ISTER_PROPERTIES)

;- Instantiation of visualization (IDLitVisPolyline) object
  visPOLY = OBJ_NEW('IDLitVisPolyline'    ,$
                    /REGISTER_PROPERTIES  ,$
                    /MANIPULATOR_TARGET   ,$ ;- that was my long lasting problem!!!
                    IMPACTS_RANGE=1       ,$
                    NAME='FOV'            ,$
                    ICON='front'          ,$
                    DESCRIPTION='FOV polyline plot')

;- add&aggregate IDLgrPolyline to the IDLitVisPolyline
  visPOLY->IDLitVisPolyline::Add,oPolyl,/AGGREGATE

;- add&aggregate IDLitVisPolyline to the iPlot
  oTool ->IDLitTool::Add, visPOLY,/AGGREGATE

;- get identifier & reference
 idvisPOLY = oTool->IDLitTool::FindIdentifiers('*FOV*', /VISUALIZATIONS,/LEAF_NODES)
 refvisPOLY = oTool->IDLitContainer::GetByIdentifier(idvisPOLY)


;- VERT_COLORS overwrite COLOR! Set it to 0 to leave COLOR free to act
success=oPlot->IDLitTool::DoSetProperty(idvisPOLY, 'VERT_COLORS',0)

refPoly_MASCS.ref->IDLITVISPOLYLINE::GetProperty, VERT_COLORS=vrt
success=oPlot->IDLitTool::DoSetProperty(idvisPOLY, 'COLOR', [255,0,0])
oPlot->IDLitTool::CommitActions

;- Set polyline color for a single shape (similar to single ROI)
colors=[80,254]
triple_col=transpose(cgColor(stringer(color,/INT), /Triple))
VERT_COLORS=lonarr(3,n_vertex*n_poly)
  for i=0,n_poly-1 do  VERT_COLORS(*,i*n_vertex:i*n_vertex+n_vertex-1)=rebin(triple
_col(*,i),3,n_vertex)

success=oPlot->IDLitTool::DoSetProperty(refPoly_MASCS.id, 'VERT_COLORS',
VERT_COLORS) & oPlot->IDLitTool::CommitActions

;-- END IDL code Area
```