Subject: Re: Preserving coordinate transformation
Posted by Wox on Tue, 17 Oct 2006 11:11:37 GMT
View Forum Message <> Reply to Message

Save the !x, !y and !map after making the plot and restore them before
you use convert_coord.

On 17 Oct 2006 03:47:17 -0700, "Sverre Solberg" <sso@nilu.no> wrote:

> I plot a map (by the MAP_SET etc routines) inside a widget_draw area
> using "/button_events". When the user is clicking on the map the
> program use the event (x and y) and the "convert_coord" to compute the
> geographical coordinates of the position. However, when making other
> plots in between, opening other draw widgets, the built-in coordinate
> transformations (device->data) changes and destroys the conversion to
> geo. coord. next time this window is clicked on. How could I store the
> coordinate transformation? If it was a simple 2D plot, without the map,
> I could just save the !x and !y variables and compute the conversion
> myself, but that dont work when there's a map projection. I then need
> to know how idl converts from device coordinates (returned by widget
> draw) to map coordinates (lat/long) and I havent been able to find out
> that. Any hints?
>
> Sverre

Subject: Re: Preserving coordinate transformation
Posted by Sverre Solberg on Tue, 17 Oct 2006 12:53:24 GMT
View Forum Message <> Reply to Message

Hm, well, that's just what I tried, still it is unclear to me how to
actually use the !x, !y and !map. I cant simply reset, like !x = <saved
!x> etc, as I guess I also need the !D which is a read-only
variable(?). Furthermore, I havent been able to dig out how/where idl
do the conversion from device to map coordinates. The documentation for
convert_coord is only describing transformation between data, device
and normal, but doesnt mention the situation when there's a map
involved. Trying to hardcode the transformation formulas described in
the manual for convert_coord (after first saving the values stored in
!x, !y and !d) doesnt seem to give the correct answer. Am I missing
some important point here?

Sverre

Wox wrote:
> Save the !x, !y and !map after making the plot and restore them before
> you use convert_coord.
>
>
> On 17 Oct 2006 03:47:17 -0700, "Sverre Solberg" <sso@nilu.no> wrote:
>
>> I plot a map (by the MAP_SET etc routines) inside a widget_draw area
>> using "/button_events". When the user is clicking on the map the
>> program use the event (x and y) and the "convert_coord" to compute the
>> geographical coordinates of the position. However, when making other
>> plots in between, opening other draw widgets, the built-in coordinate
>> transformations (device->data) changes and destroys the conversion to
>> geo. coord. next time this window is clicked on. How could I store the
>> coordinate transformation? If it was a simple 2D plot, without the map,
>> I could just save the !x and !y variables and compute the conversion
>> myself, but that dont work when there's a map projection. I then need
>> to know how idl converts from device coordinates (returned by widget
>> draw) to map coordinates (lat/long) and I havent been able to find out
>> that. Any hints?
>>
>> Sverre

Subject: Re: Preserving coordinate transformation
Posted by David Fanning on Tue, 17 Oct 2006 13:12:44 GMT
View Forum Message <> Reply to Message

Sverre Solberg writes:

> Hm, well, that's just what I tried, still it is unclear to me how to
> actually use the !x, !y and !map. I cant simply reset, like !x = <saved
> !x> etc, as I guess I also need the !D which is a read-only
> variable(?). Furthermore, I havent been able to dig out how/where idl
> do the conversion from device to map coordinates. The documentation for
> convert_coord is only describing transformation between data, device
> and normal, but doesnt mention the situation when there's a map
> involved. Trying to hardcode the transformation formulas described in
> the manual for convert_coord (after first saving the values stored in
> !x, !y and !d) doesnt seem to give the correct answer. Am I missing
> some important point here?

What's usually missing in most of the widget programs
I look at is a failure to make *this* window the current
graphics window! A WSET will probably cure your !D problems
(which you don't need to bother with, by the way).

Do a WSET to the proper window, restore !X, !Y, and !MAP,
and CONVERT_COORD will work like a champ. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Preserving coordinate transformation
Posted by Sverre Solberg on Tue, 17 Oct 2006 13:32:09 GMT
View Forum Message <> Reply to Message

Thanks, you were right! A bit silly this, as I had specified the window
index with wset, but another bug made it wrong. Anyway, it works now.
And thanks for the help, I wasn't aware that wset reestablished !D.

David Fanning wrote:
> Sverre Solberg writes:
>
>> Hm, well, that's just what I tried, still it is unclear to me how to
>> actually use the !x, !y and !map. I cant simply reset, like !x = <saved
>> !x> etc, as I guess I also need the !D which is a read-only
>> variable(?). Furthermore, I havent been able to dig out how/where idl
>> do the conversion from device to map coordinates. The documentation for
>> convert_coord is only describing transformation between data, device
>> and normal, but doesnt mention the situation when there's a map
>> involved. Trying to hardcode the transformation formulas described in
>> the manual for convert_coord (after first saving the values stored in
>> !x, !y and !d) doesnt seem to give the correct answer. Am I missing
>> some important point here?
>
> What's usually missing in most of the widget programs
> I look at is a failure to make *this* window the current
> graphics window! A WSET will probably cure your !D problems
> (which you don't need to bother with, by the way).
>
> Do a WSET to the proper window, restore !X, !Y, and !MAP,
> and CONVERT_COORD will work like a champ. :-)
>
> Cheers,
>
> David

> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Preserving coordinate transformation
Posted by David Fanning on Tue, 17 Oct 2006 14:14:20 GMT
View Forum Message <> Reply to Message

Sverre Solberg writes:

> And thanks for the help, I wasn't aware that wset reestablished !D.

I have the advantage of having recently spent several glorious days in the thrall of coordinate conversion mechanisms. :-)

I did learn something else of interest in the three excruciating days I spent considering all of this. I was working with a window that had several images in it. Each image has associated with it its own data coordinate system. My draw widget is able to "select" which image I clicked in, and tells that image to establish its data coordinate system.

Now, I usually need to convert the window location (event.x and event.y) into the data coordinate system of the image. No problem with CONVERT_COORD, in the manor I outlined previously:

```
d = Convert_Coord(event.x, event.y, /Device, /To_Data)
```

BUT, on occasion I need to convert a *distance* in device coordinates into a distance in data coordinates. For example, if I want to draw a station plot at the location where I clicked, but I want to add a station name just *above* the location, I may want to move the text 10 pixels toward the top of the image.

Naively, I was doing the conversion like this:

```
d = Convert_Coord(10, 10, /Device, /To_Data)
```

Then adding the "length" to the y data coordinate:

```
y_data = y_data + d[1]
```

This worked great with some images, and not at all with others! It turns out it worked great when I had one image in the window and the image coordinate system overlapped more or less with the device coordinate system. (Or, when the image I selected was near the origin of the device coordinate system.) But if an image was far away from the origin of the device coordinate system my text could be WAY off the mark.

I realized, after much futzing around and thinking through an unhappy dinner with my wife ("Are you paying attention to what I'm talking about!?") that this was the wrong way to calculate a distance or length in my data coordinate system.

What I needed to know, was how many data coordinate "units" were in 10 "units" of device coordinates. This depended on my data coordinate range and is something CONVERT_COORD cannot compute for me. In fact, I realized I had to do the conversion myself, like this:

```
d = Abs((yrange[1] - yrange[0])) / !D.Y_Size * 10
y_data = y_data + d
```

Clearing up this little misunderstanding solved a LOT of problems for me. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")
