Subject: scopes
Posted by spluque on Fri, 20 Sep 2013 21:28:04 GMT
View Forum Message <> Reply to Message

Hi,

Using the following script to convert calendar date to day of year (test.pro):

```
FUNCTION calendar2doy, year, month, day
  jd=julday(month, day, year)
  caldat, jd, Null, Null, year
  doy=string(jd - julday(12, 31, year - 1), format='(i03)')
  RETURN, doy
END

PRO TEST
  year=2011
  mon=10
  day=15
  DOY=calendar2doy(year, mon, day)
  RETURN
END
```

I expected the variable year in the TEST procedure to remain as defined (the long integer 2011), but this is what I see after calling the call to calendar2doy with a breakpoint at the RETURN line:

```
IDL> .run "test.pro"
% Compiled module: CALENDAR2DOY.
% Compiled module: TEST.
IDL> breakpoint,'test.pro',14
IDL> test
% Compiled module: JULDAY.
% Compiled module: CALDAT.
% Breakpoint at: TEST             14 test.pro
IDL> print, year
       2012
```

What am I missing?

Cheers,
Seb

Subject: Re: scopes
Posted by David Fanning on Fri, 20 Sep 2013 21:44:22 GMT
View Forum Message <> Reply to Message

spluque@gmail.com writes:

>
> Hi,
>
> Using the following script to convert calendar date to day of year (test.pro):
>
> FUNCTION calendar2doy, year, month, day
> jd=julday(month, day, year)
> caldat, jd, Null, Null, year
> doy=string(jd - julday(12, 31, year - 1), format='(i03)')
> RETURN, doy
> END
>
> PRO TEST
> year=2011
> mon=10
> day=15
> DOY=calendar2doy(year, mon, day)
> RETURN
> END
>
>
> I expected the variable year in the TEST procedure to remain as defined (the long integer 2011), but this is what I see after calling the call to calendar2doy with a breakpoint at the RETURN line:
>
> IDL> .run "test.pro"
> % Compiled module: CALENDAR2DOY.
> % Compiled module: TEST.
> IDL> breakpoint,'test.pro',14
> IDL> test
> % Compiled module: JULDAY.
> % Compiled module: CALDAT.
> % Breakpoint at: TEST 14 test.pro
> IDL> print, year
> 2012
>
> What am I missing?

Ah, yes, I've seen this before. It's weird. :-)

The problem comes about in the way you are calling CalDat:

  caldat, jd, Null, Null, year

You are using the same variable for the day and month. This causes
CalDat great confusion! If you use different variables, you will get

what you expect.

  caldat, jd, Null1, Null2, year

It must be something about variables getting updated in a particular
sequence or something. I don't understand exactly what is happening, but
I remember struggling for hours with exactly this thing one time. ;-)

Cheers,

David

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

Subject: Re: scopes
Posted by spluque on Fri, 20 Sep 2013 21:52:19 GMT
View Forum Message <> Reply to Message

It looks like those NULL arguments for caldat are causing problems. If I supply some variable (which I don't need at all), then year keeps its value. This is very weird, nonetheless, and it would be good to know what is going on.

Seb

Subject: Re: scopes
Posted by spluque on Fri, 20 Sep 2013 22:00:27 GMT
View Forum Message <> Reply to Message

Thanks, David, I followed up at about the same time, just after I played supplying some variable names for month and day. It's mildly annoying to have to do this since one may have no use for them, cluttering the environment.

Seb

Subject: Re: scopes
Posted by SonicKenking on Mon, 14 Oct 2013 03:20:53 GMT
View Forum Message <> Reply to Message

On Saturday, September 21, 2013 8:00:27 AM UTC+10, spl...@gmail.com wrote:

> Thanks, David, I followed up at about the same time, just after I played supplying some variable names for month and day. It's mildly annoying to have to do this since one may have no use for them, cluttering the environment.
>
>
>
> Seb

You can do it with the new !NULL variable if you have IDL 8.0+, like this
caldat, jd, !null, !null, year

Correct result. No dummy variables.

Subject: Re: scopes
Posted by Matthew Argall on Tue, 15 Oct 2013 15:23:38 GMT
View Forum Message <> Reply to Message

Variables are passed by reference in IDL. If two output positional parameters reference the same variable name, you are going to have a bad time.

Try this

```
---------------------------------
pro test_input_params, a, b
   a = 5
   b = 4
   help, a, b
end

test_input_params, null, null
---------------------------------
```

It shows that a = a = 4 and that b = a = 4 inside the test program. If I were to later do something with a and b independently, the results would not be independent. I assume that CalDat uses the "month" and "day" parameters when calculating "year", which is why your output is funky.

Subject: Re: scopes
Posted by Matthew Argall on Tue, 15 Oct 2013 15:41:09 GMT
View Forum Message <> Reply to Message

I guess a better example would be this:

```
---------------------------------------
function test_input_params, a, b
   a = 5
```

```
    b = 4
    return, a + b
end

IDL> print, test_input_params(null, null)
       8
```