Subject: System Variable Behavior Changed in IDL 6.0

Posted by David Fanning on Thu, 11 Mar 2004 05:32:23 GMT

View Forum Message <> Reply to Message

Folks,

I wanted to display some images today in a !P.MULTI situation.
It used to be that my TVIMAGE program could handle !P.MULTI
events automatically. (Don't try this at home with TV.) But
this appeared to be broken in IDL 6.0. No !P.MULTI behavior
at all!

I traced the problem to a change in how system variables
behave in IDL 6.0. Here is a sequence of plots in IDL 5.6.
Notice now the !X and !Y window fields update the plot
location as you go.

```
IDL> Print, !Version
{ x86 Win32 Windows Microsoft Windows 5.6 Oct 26 2002     32     64}
IDL> !P.Multi=[0,2,3]
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
   0.0823579    0.703035    0.475299    0.981823
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
    0.582358    0.703035    0.975299    0.981823
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
   0.0823579    0.369702    0.475299    0.648490
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
    0.582358    0.369702    0.975299    0.648490
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
   0.0823579    0.0363686    0.475299    0.315157
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
    0.582358    0.0363686    0.975299    0.315157
```

Here is the same sequence of commands in IDL 6.0. No updating of
the system variables. :-(

```
IDL> print, !Version
{ x86 Win32 Windows Microsoft Windows 6.0 Jun 27 2003     32     64}
IDL> !P.Multi=[0,2,3]
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
   0.150000    0.125000    0.950000    0.950000
```

```
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
   0.150000    0.125000    0.950000    0.950000
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
   0.150000    0.125000    0.950000    0.950000
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
   0.150000    0.125000    0.950000    0.950000
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
   0.150000    0.125000    0.950000    0.950000
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
   0.150000    0.125000    0.950000    0.950000
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
   0.150000    0.125000    0.950000    0.950000
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
   0.150000    0.125000    0.950000    0.950000
```

Is this a Windows thing? Collecting data for a call to RSI. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: System Variable Behavior Changed in IDL 6.0
Posted by Dick Jackson on Thu, 11 Mar 2004 21:01:22 GMT
View Forum Message <> Reply to Message

Hi David,

"David Fanning" <david@dfanning.com> wrote in message
news:MPG.1ab9a7d49f84162c9896f7@news.frii.com...
> Folks,
>
> I wanted to display some images today in a !P.MULTI situation.
> It used to be that my TVIMAGE program could handle !P.MULTI
> events automatically. (Don't try this at home with TV.) But

> this appeared to be broken in IDL 6.0. No !P.MULTI behavior
> at all!
>
> I traced the problem to a change in how system variables
> behave in IDL 6.0. Here is a sequence of plots in IDL 5.6.
> Notice now the !X and !Y window fields update the plot
> location as you go.
>
> [...]
>
> Here is the same sequence of commands in IDL 6.0. No updating of
> the system variables. :-(
>
> IDL> print, !Version
> { x86 Win32 Windows Microsoft Windows 6.0 Jun 27 2003   32   64}
>
> [...]

That's a bummer! Good news and bad news: mine on Windows is working
OK...

```
IDL> Print, !Version
{ x86 Win32 Windows Microsoft Windows 6.0 Jun 27 2003    32    64}
IDL> !P.Multi=[0,2,3]
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
   0.0562550    0.706672    0.483130    0.980005
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
    0.556255    0.706672    0.983130    0.980005
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
   0.0562550    0.373338    0.483130    0.646672
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
    0.556255    0.373338    0.983130    0.646672
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
   0.0562550   0.0400050    0.483130    0.313338
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
    0.556255   0.0400050    0.983130    0.313338
```

I did this at startup to the 'WIN' device, what device were you drawing
to? Let us know what you find out.

Cheers,
--

-Dick

Dick Jackson   /   dick@d-jackson.com
D-Jackson Software Consulting /   http://www.d-jackson.com
Calgary, Alberta, Canada   / +1-403-242-7398 / Fax: 241-7392

Subject: Re: System Variable Behavior Changed in IDL 6.0
Posted by David Fanning on Thu, 11 Mar 2004 21:46:40 GMT
View Forum Message <> Reply to Message

Dick Jackson writes:

> That's a bummer! Good news and bad news: mine on Windows is working
> OK...
>
> IDL> Print, !Version
> { x86 Win32 Windows Microsoft Windows 6.0 Jun 27 2003   32   64}
> IDL> !P.Multi=[0,2,3]
> IDL> plot, findgen(11)
> IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
>   0.0562550   0.706672   0.483130   0.980005

Oh, oh. One of these files (there are more than 100) in
this project from hell must be setting a system variable,
because if I run the program in a new IDL session, I have
no problems and IDL acts like it is suppose to. Now, to
track down the offending file...

Cheers,

David

P.S. Did I mention I love objects?

--
David Fanning, Ph.D.
Fanning Software Consulting
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: System Variable Behavior Changed in IDL 6.0
Posted by mperrin+news on Thu, 11 Mar 2004 21:52:03 GMT
View Forum Message <> Reply to Message

Michael Wallace <mwallace.removethismunge@swri.edu.invalid> wrote:
>> just create two soft links:

>>
>> ln -s /usr/local/rsi/idl_5.5/bin/idl /usr/local/bin/idl55
>> ln -s /usr/local/rsi/idl_6.0/bin/idl /usr/local/bin/idl60
>>
>> and use the idl55 and idl60 commands (idl is a script itself, which sets
>> the appropriate env. variables).
>
> In this case, you still have the issue of what IDL_PATH is set to. If
> left unset, the soft links solution you suggest will work. IDL will
> automagically determine the correct lib directory to use.
>
> However, almost all of us have to explicitly set IDL_PATH in order to
> include our own libraries and other procedures. Because of this we are
> forced to choose between the $RSI_DIR/idl_5.5/lib directory and the
> $RSI_DIR/idl_6.0/lib directories. And you can't set IDL_PATH to include
> both directories since one will clobber the other.
>
> So, with that in mind, the better solution is to symlink the IDL
> directories rather than the executables. By using a symlink that points
> to idl_5.5 or idl_6.0 or other directory, we don't have the IDL_PATH
> issue because we just have to only include $RSI_DIR/idl/lib in IDL_PATH.
>   So, all we have to do is change the symlink and we're ready to go.

Alternatively, one can just use environment variables:

----
setenv IDL_DIR /usr/local/rsi/idl_6.0
#setenv IDL_DIR /usr/local/rsi/idl_5.6

setenv LM_LICENSE_FILE $IDL_DIR/license/license.dat
setenv IDL_PATH +$IDL_DIR/lib:+$HOME/idl

Put this in your .tcshrc (or equivalent) and all you have to do is change
which of the first two lines is uncommented and then start a new shell.

 - Marshall

Subject: Re: System Variable Behavior Changed in IDL 6.0
Posted by Michael Wallace on Thu, 11 Mar 2004 22:33:59 GMT
View Forum Message <> Reply to Message

> Alternatively, one can just use environment variables:

I knew there was an easier way!! Maybe trying to read the External
Development Guide today has made me thing that everything must be hard!

```
>  -------
>  setenv IDL_DIR /usr/local/rsi/idl_6.0
>  #setenv IDL_DIR /usr/local/rsi/idl_5.6
>
>  setenv LM_LICENSE_FILE $IDL_DIR/license/license.dat
>  setenv IDL_PATH +$IDL_DIR/lib:+$HOME/idl
>  -------
```

Typically, the license directory is under $RSI_DIR, not $IDL_DIR. Also,
make sure that your $PATH is updated also so that the correct
executables are called.

For csh and equivalent:
setenv RSI_DIR /usr/local/rsi
setenv IDL_DIR $RSI_DIR/idl_6.0 # or idl_5.6

setenv LM_LICENSE_FILE $RSI_DIR/license/license.dat
setenv IDL_PATH +$IDL_DIR/lib
setenv PATH $PATH:$IDL_DIR/bin

Subject: Re: System Variable Behavior Changed in IDL 6.0
Posted by David Fanning on Thu, 11 Mar 2004 23:10:47 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> Here is the same sequence of commands in IDL 6.0. No updating of
> the system variables. :-(
>
> IDL> print, !Version
> { x86 Win32 Windows Microsoft Windows 6.0 Jun 27 2003   32   64}
> IDL> !P.Multi=[0,2,3]
> IDL> plot, findgen(11)
> IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
>   0.150000   0.125000   0.950000   0.950000
> IDL> plot, findgen(11)
> IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
>   0.150000   0.125000   0.950000   0.950000

Ok, this turns out to be my fault, sorta. :-)

This kind of problem happens when you have !P.Position
set to some value:

   !P.Position = [0.15, 0.125, 0.95, 0.95]

If this is the case, !P.MULTI gets confused, etc., etc.

I happen to know this.

In my case, I inherited some code that sets !P.Position,
and all I am trying to do it set is back to its *default*
value. But, alas, there is no way, as far as I know, to
set !P.Position to its default value!!

Let's start at the beginning. What *is* its default value?
We start a fresh IDL session to find out:

```
  IDL> Help, !P.Position
  <Expression>    FLOAT    = Array[4]
  IDL> Print, !P.Position
      0.000000    0.000000    0.000000    0.000000
```

Ok, so now we change it to something:

```
  IDL> !P.Position = [0.15, 0.125, 0.95, 0.95]
```

And we are done with it, so we change it back:

```
  IDL> !P.Position = FltArr(4)
```

All well and good, until we go to plot something:

```
  IDL> Plot, findgen(11)
```

Oh, oh! :-(

IDL is apparently going to make me set it to *something*
to get my plots to look right again, but if I do, all
my !P.MULTI plots are shot to hell.

Anybody have any bright ideas?

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: System Variable Behavior Changed in IDL 6.0
Posted by MKatz843 on Fri, 12 Mar 2004 00:33:01 GMT

Also, works as expected on Mac

```
IDL> Print, !Version
{ ppc darwin unix Mac OS X 6.0 Jun 27 2003     32     32}
IDL> !P.Multi=[0,2,3]
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
   0.0468800    0.705734    0.485943    0.980474
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
    0.546880    0.705734    0.985942    0.980474
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
   0.0468800    0.372401    0.485943    0.647140
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
    0.546880    0.372401    0.985942    0.647140
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
   0.0468800    0.0390675    0.485943    0.313807
IDL> plot, findgen(11)
IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
    0.546880    0.0390675    0.985942    0.313807
```

Subject: Re: System Variable Behavior Changed in IDL 6.0
Posted by mperrin+news on Fri, 12 Mar 2004 05:48:16 GMT

Michael Wallace <mwallace.removethismunge@swri.edu.invalid> wrote:
>> Alternatively, one can just use environment variables:
>
> I knew there was an easier way!! Maybe trying to read the External
> Development Guide today has made me thing that everything must be hard!
>
>> -------
>> setenv IDL_DIR /usr/local/rsi/idl_6.0
>> #setenv IDL_DIR /usr/local/rsi/idl_5.6
>>
>> setenv LM_LICENSE_FILE $IDL_DIR/license/license.dat
>> setenv IDL_PATH +$IDL_DIR/lib:+$HOME/idl
>> -------
>
> Typically, the license directory is under $RSI_DIR, not $IDL_DIR.

Ah, OK, that must be a peculiarity of my site's installation. We've got

difference license servers running for the various different versions.

> Also,
> make sure that your $PATH is updated also so that the correct
> executables are called.

Here's where it gets really slick. If you examine
/usr/local/rsi/idl_6.0/bin/idl, you'll see that it's actually a shell
script that uses $IDL_DIR to find and then execute something like
/usr/local/rsi/idl_6.0/bin/bin.solaris2.sparc/idl

Thus _no matter which */bin/idl is in your $PATH, if $IDL_DIR is set
it will find and run the correct binary_!

I've got all of IDL 5.2 through 6.0 installed and this trick works
to switch between all of them, with the $IDL_DIR environment variable
as the only single thing that changes.

 - Marshall

Subject: Re: System Variable Behavior Changed in IDL 6.0
Posted by Pepijn Kenter on Fri, 12 Mar 2004 08:51:54 GMT
View Forum Message <> Reply to Message

Marshall Perrin wrote:
> Michael Wallace <mwallace.removethismunge@swri.edu.invalid> wrote:
>
>>> Alternatively, one can just use environment variables:
>>
>> I knew there was an easier way!! Maybe trying to read the External
>> Development Guide today has made me thing that everything must be hard!
>>
>>
>>> -------
>>> setenv IDL_DIR /usr/local/rsi/idl_6.0
>>> #setenv IDL_DIR /usr/local/rsi/idl_5.6
>>>
>>> setenv LM_LICENSE_FILE $IDL_DIR/license/license.dat
>>> setenv IDL_PATH +$IDL_DIR/lib:+$HOME/idl
>>> -------
>>
>> Typically, the license directory is under $RSI_DIR, not $IDL_DIR.
>
>
> Ah, OK, that must be a peculiarity of my site's installation. We've got
> difference license servers running for the various different versions.
>

>
>> Also,
>> make sure that your $PATH is updated also so that the correct
>> executables are called.
>
>
> Here's where it gets really slick. If you examine
> /usr/local/rsi/idl_6.0/bin/idl, you'll see that it's actually a shell
> script that uses $IDL_DIR to find and then execute something like
> /usr/local/rsi/idl_6.0/bin/bin.solaris2.sparc/idl
>
> Thus _no matter which */bin/idl is in your $PATH, if $IDL_DIR is set
> it will find and run the correct binary_!
>
> I've got all of IDL 5.2 through 6.0 installed and this trick works
> to switch between all of them, with the $IDL_DIR environment variable
> as the only single thing that changes.
>
>   - Marshall

My machine has the soft links pointing to the IDL directories and all
environment variables and aliases are using this softlink. This way all
paths are set correctly.

I realize now that my main problem with this solution was not the hassle
of creating a softlink every time I wanted to change versions, but that
I had to make sure the one or two other users on my machine are not
currently working in IDL (don't worry, I did). Setting the correct
environment variables in a script (or my .bashrc) will solve this
problem. Also the other users, who don't have the root password, can
then change versions if they like.

Thank you all for you input.

Best regards, Pepijn.

Subject: Re: System Variable Behavior Changed in IDL 6.0
Posted by westrick85 on Fri, 12 Mar 2004 13:18:21 GMT
View Forum Message <> Reply to Message

Michael Wallace <mwallace.removethismunge@swri.edu.invalid> wrote in message
news:<1051l7jad48891b@corp.supernews.com>...
> In this case, you still have the issue of what IDL_PATH is set to. If
> left unset, the soft links solution you suggest will work. IDL will
> automagically determine the correct lib directory to use.

Try using <IDL_DEFAULT> inside your IDL_PATH environment variable like

this:

csh:
setenv IDL_PATH "<IDL_DEFAULT>:+/foo/idllib"

bash:
export IDL_PATH="<IDL_DEFAULT>:+/foo/idllib"

IDL expands IDL_DEFAULT into the right lib directory based on which
version of IDL you are running. This way you don't have to change
IDL_PATH if you switch to running a different version of IDL.

Corey

Subject: Re: System Variable Behavior Changed in IDL 6.0
Posted by Stein Vidar Hagfors H[2] on Fri, 12 Mar 2004 17:17:35 GMT
View Forum Message <> Reply to Message

David Fanning <david@dfanning.com> writes:

> Dick Jackson writes:
>
>> That's a bummer! Good news and bad news: mine on Windows is working
>> OK...
>>
>> IDL> Print, !Version
>> { x86 Win32 Windows Microsoft Windows 6.0 Jun 27 2003   32   64}
>> IDL> !P.Multi=[0,2,3]
>> IDL> plot, findgen(11)
>> IDL> print, !X.Window[0], !Y.Window[0], !X.Window[1], !Y.Window[1]
>>   0.0562550   0.706672   0.483130   0.980005
>
> Oh, oh. One of these files (there are more than 100) in
> this project from hell must be setting a system variable,
> because if I run the program in a new IDL session, I have
> no problems and IDL acts like it is suppose to. Now, to
> track down the offending file...

See http://idlastro.gsfc.nasa.gov/ftp/pro/plot/cleanplot.pro
(i.e. in the idl astronomy library)

It's a simple matter of "good plotting hygiene" :-)

Recommended usage: Any time you make the assumption that nobody has
changed the !P, !X, !Y, !Z variables and are proved wrong!

--

-- -------------
Stein Vidar Hagfors Haugan
ESA SOHO SOC/European Space Agency Science Operations Coordinator for SOHO

NASA Goddard Space Flight Center,   Tel.: 1-301-286-9028
Mail Code 682.3, Bld. 26, Room G-1, Cell: 1-240-354-6066
Greenbelt, Maryland 20771, USA.   Fax: 1-301-286-0264
-- -------------

Subject: Re: System Variable Behavior Changed in IDL 6.0
Posted by David Fanning on Fri, 12 Mar 2004 17:58:17 GMT
View Forum Message <> Reply to Message

Stein Vidar Hagfors Haugan writes:

> See http://idlastro.gsfc.nasa.gov/ftp/pro/plot/cleanplot.pro
> (i.e. in the idl astronomy library)
>
> It's a simple matter of "good plotting hygiene" :-)
>
> Recommended usage: Any time you make the assumption that nobody has
> changed the !P, !X, !Y, !Z variables and are proved wrong!

Well, this story just gets stranger and stranger.

Here is the *real* problem. Like almost all system variables,
you ought to be able to set the !P.Position system variable
to its default value by doing this:

  !P.Position = 0

But, for some reason, in IDL 5.6 (Windows 2000 Pro), you
couldn't do this. It would make the position=[0,0,1,1],
instead of the "default", whatever that is.

So I came up with some elaborate scheme to get around this
and keep my programs working. But this has been fixed in
IDL 6.0, so my "problem" is that my hair-brained scheme
for IDL 5.6 is breaking everything in IDL 6.0.

Welcome to the wonderful world of backward compatibility. :-(

Cheers,

David

P.S. Let's just say I've entered into negotiations with Craig

to get ahold of his old IDL 4.0. At least there you know where
you are!

--
David Fanning, Ph.D.
Fanning Software Consulting
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: System Variable Behavior Changed in IDL 6.0
Posted by Craig Markwardt on Fri, 12 Mar 2004 22:26:24 GMT

View Forum Message <> Reply to Message

David Fanning <david@dfanning.com> writes:
>
> P.S. Let's just say I've entered into negotiations with Craig
> to get ahold of his old IDL 4.0. At least there you know where
> you are!

I've got plenty of spare round parentheses around if you need them.

Craig