
Subject: Re: IDL 5.3 serious problem: save files sneakily restored
Posted by [Martin Schultz](#) on Fri, 25 Aug 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

>
> Vinay L. Kashyap (kashyap@head-cfa.harvard.edu) writes:
>
>> IDL 5.3 appears to have a major hull breach.
>>
>> If there is an idl save file named "<whatever>.sav" in the current
>> directory,
>> the contents of that file get restored whenever <whatever> is referenced as
>> a function within IDL.
>>
>> Comments?
>
> Uh, I'm not sure it's a hull breach. If I understand
> you correctly this has been a behavior of IDL for a
> long, long time.
>
> If you have a piece of code like this:
>
> IDL> a = junker()
>
> IDL assumes that JUNKER is a function. Many people
> think IDL goes looking in the !Path for a file junker.pro,
> but in fact it looks for a file junker.sav *first*. This
> is so that if you have compiled the file junker.pro and
> saved it:
>

This is at least contrary to the online help of 5.3:

Automatic Execution

When a file is specified by typing only the filename at the IDL prompt, IDL searches the current directory for filename.pro (where filename is the file specified) and then for filename.sav. If no file is found in the current directory, IDL searches in the same way in each directory specified by !PATH. If a file is found, IDL automatically compiles the contents and executes any functions or procedures that have the same name as the file specified (excluding the suffix).

So, at least theoretically, pro should get executed before sav. And this makes all the sense in the world, because otherwise it would be hard for authors of compiled files to develop them -

they would have to delete the sav file each time they modify the pro file and want to test changes.

Cheers,
Martin

> P.S. Of course, these would have to be LONG integers.
Oh so true! This cost me about an hour yesterday to figure out why my grid generation program (gaussian grid) would fail for resolutions finer than T106 (320x160 points), even though it was an almost 1:1 conversion from a fortran program which works for all resolutions. And, then, finally, I realized that
(8.*nlat^2)
would not yield what I had intended ...

```
IDL> nlat=[1,2,4,8,16,32,64,128,256]
IDL> print,8.*nlat^2
      8.00000      32.0000      128.000      512.000      2048.00
    8192.00     32768.0     131072.      0.00000
```

—

[[Dr. Martin Schultz Max-Planck-Institut fuer Meteorologie
[[
[[Bundesstr. 55, 20146 Hamburg
[[
[[phone: +49 40 41173-308
[[
[[fax: +49 40 41173-298
[[
[[martin.schultz@dkrz.de
[[
[[