Subject: Re: Implied print and namespace
Posted by chris_torrence@NOSPAM on Tue, 10 Jun 2014 13:40:45 GMT
View Forum Message <> Reply to Message

On Tuesday, June 10, 2014 3:10:00 AM UTC-6, Fabien wrote:
> Hi everyone,
>
>
>
> this is probably not new to the IDL team, but implied print does not
>
> work all the time:
>
>
>
> IDL> image = 1.
>
> IDL> image
>
> % Compiled module: IMAGE.
>
> % Attempt to call undefined procedure/function: 'IMAGE'.
>
> % Execution halted at: $MAIN$
>
>
>
> Ok, never call your variable like an IDL function. But what about "t"?
>
>
>
> IDL> t = 1.
>
> IDL> print, t
>
> 1.00000
>
> IDL> t
>
> % Restored file: T.
>
> % Attempt to call undefined procedure/function: 'T'.
>
> % Execution halted at: $MAIN$
>
>
>
> Uh?

>
>
>
> Cheers,
>
>
>
> Fabien

Hi Fabien,

Well, there must be a "t.sav" somewhere on your IDL path. Try file_which (with implied print):
IDL> file_which('t.sav')

Anyway, IDL will never do implied print if the variable name matches a current routine name - otherwise we would have broken backwards compatibility, which would be very bad.

There are also a few other cases where implied print will not work. The main one is nested structures. For example:

```
IDL> a = {b:{c:5}}
IDL> ; the following works fine
IDL> a.b
{
   "C": 5
}
IDL> ; this will throw an error
IDL> a.b.c
% Object reference type required in this context: <STRUCT    Array[1]>.
% Execution halted at: $MAIN$
```

This is because IDL cannot determine (at compile time) whether "a.b.c" is a structure dereference or a method call. It would be possible to do this at run time, but that might slow down the interpreter, and didn't seem worth the effort.

Hope this helps!

Cheers,
Chris
ExelisVIS