Subject: Re: Another small V8.0 bug
Posted by penteado on Mon, 26 Jul 2010 17:59:24 GMT
View Forum Message <> Reply to Message

On Jul 26, 2:12 pm, wlandsman <wlands...@gmail.com> wrote:
> I have found another bug in V8.0, at least for users who still have
> round parenthesis used for indices lurking around in their code.
> Like Mike Potter's example, it is not easily repeatable, and for
> example sometimes only occurs after compile the program a second
> time. And because it occurs in a fairly large program, I have not
> yet isolated it into a simple test program. But I can illustrate
> the problem after placing a STOP statement
>
> % Stop encountered: SHOWDB 72 /home/landsman/uvot/bpm16274/
> showdb.pro
> IDL> help,list
> LIST LONG = Array[1]
> IDL> print,list[0]
> 183
> IDL> print,list(0)
> 0
> IDL> help,list(0)
> <Expression> LIST <ID=22 NELEMENTS=1>
> IDL> print,list(0) LE 0
> % Unable to convert variable to type object reference.
> % Execution halted at: SHOWDB 72 /home/landsman/uvot/
> bpm16274/showdb.pro
> % $MAIN$
>
> IDL> print,!version
> { x86 linux unix linux 8.0 Jun 18 2010 32 64}

Do you mean that this does not happen every time? To me this seems to
be the expected behavior. The line

print,list(0)

Is creating a list (which is an object), containing one element, and
printing it. The same with the use of help.

>
> So IDL seems confused as to whether 'list' is a variable or an
> object. (The code is all imperative statements with no object
> syntax). Note that this differs from the long-standing variable/
> function ambiguity that can occur when using the () syntax for
> indexing. --Wayne

Actually, this is the same old function/variable ambiguity. In this case, between the array called list, and the function called list() - which just happens to be the function that creates a list object. The only new things here is that objects can be instantiated with the syntax object=class(), which is equivalent to object=object_new('class'), and that since there is a new builtin class called 'list', that line is being interpreted as a function call.

Perhaps the variability you report is from the context, which determines whether () are interpreted as indexing or function call, since that depends on which variables are currently defined, and which functions are currently compiled.
