Subject: David: for your book? Posted by Martin Schultz on Tue, 15 Aug 2000 07:00:00 GMT View Forum Message <> Reply to Message Hi, Testing IDL's capabilities is always dangerous: you might get more than you ask for! What would you expect to get out of the following: test = ['dir1/', 'directory2/', 'another_directory/'] print,strmid(test,0,strlen(test)-1) ?? Well, I at least had expected to get the directory names without the trailing '/'. But. no! help, strmid(test,0,strlen(test)-1) reveals that the result is a 3x3 array!! To print it nicely, use a format statement: print,strmid(test,0,strlen(test)-1),format='(3(3(">",A,"<"),/))' So, to achieve what I had in mind I either have to loop or extract the matrix diagonale! Does this make any sense, or is IDL simply overachieving here? Cheers, Martin [[Dr. Martin Schultz Max-Planck-Institut fuer Meteorologie П Bundesstr. 55, 20146 Hamburg [[phone: +49 40 41173-308 \prod \prod fax: +49 40 41173-298 [[

[[

Subject: Re: David: for your book?
Posted by Craig Markwardt on Tue, 15 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz <martin.schultz@dkrz.de> writes:

[[martin.schultz@dkrz.de

- > As an aside, I recently found another problem in terms of backward
- > compability (I am
- > sure something similar was already mentioned in this newsgroup, but this
- > was the first
- > time I had this problem):
- > I tried to write a routine that takes advantage of the new StrMatch
- > and StRegex
- > functions in 5.3. In order to ensure backward compatibility, I enclosed
- > these function calls
- > with
- > IF !Version.Release GE 5.3 THEN BEGIN
- > ENDIF

>

- > However, a colleague tells me: no chance! The program will report a
- > syntax error when run in
- > IDL 5.2! So, the problem is that IDL looks for a function of this name
- > during compile stage
- > and doesn't care if it will ever be used.

. . .

There is a way to do this without defining an external function, by using FORWARD_FUNCTION properly:

FORWARD_FUNCTION stregex, strmatch
IF !Version.Release GE 5.3 THEN BEGIN
foo = stregex(bar)
sm = strmatch(eg)
ENDIF

This will make sure that IDL will always recognize strmatch & stregex as functions during the compilation stage, and there will be no syntax errors in any IDL version. It is harmless in IDL 5.3. If you really want the warning message, then by all means go ahead and make your own strmatch definitions et al.

Have fun, Craig	
•	craigmnet@cow.physics.wisc.edu Remove "net" for better response

Subject: Re: David: for your book?

Posted by davidf on Tue, 15 Aug 2000 07:00:00 GMT

Martin Schultz (martin.schultz@dkrz.de) writes:

- > So, the problem is that IDL looks for a function of this name
- > during compile stage and doesn't care if it will ever be used.

Yes. I thought I was going crazy recently, because I don't remember IDL ever complaining about not being able to find a function until run-time. But clearly it is looking for this functions during compile time.

As you suggest, this makes it awfully hard to write version-independent code. Perhaps it is done as an impetus to upgrade to the latest version of IDL. :-)

Cheers,

David

David Fanning, Ph.D.

Fanning Software Consulting

Phone: 970-221-0438 E-Mail: davidf@dfanning.com

Coyote's Guide to IDL Programming: http://www.dfanning.com/

Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: David: for your book? Posted by Martin Schultz on Tue, 15 Aug 2000 07:00:00 GMT View Forum Message <> Reply to Message

```
Craig Markwardt wrote:
```

```
> Martin Schultz <martin.schultz@dkrz.de> writes:
>> Hi,
>>
      Testing IDL's capabilities is always dangerous: you might get more
>>
>> than you ask for!
>>
>> What would you expect to get out of the following:
>> test = [ 'dir1/', 'directory2/', 'another_directory/' ]
>> print,strmid(test,0,strlen(test)-1)
> Hmm, I got an error message!
> IDL> test = [ 'dir1/', 'directory2/', 'another_directory/' ]
```

- > IDL> print,strmid(test,0,strlen(test)-1)
- > % STRMID: Expression must be a scalar in this context: <LONG Array>.
- > % Execution halted at: \$MAIN\$

>

- > IDL> print, !version
- > { alpha OSF unix 5.2 Oct 30 1998}

>

> Is this something new in IDL 5.3?

>

Apparently so! What I get as a result is a 3x3 matrix containing in the first column

all directory namews truncated to a length of 4, in the second column 10, and in the

third column StrLen(test[2]-1). But I am glad you point this out.

As an aside, I recently found another problem in terms of backward compability (I am

sure something similar was already mentioned in this newsgroup, but this was the first

time I had this problem):

I tried to write a routine that takes advantage of the new StrMatch and StRegex

functions in 5.3. In order to ensure backward compatibility, I enclosed these function calls

with

IF !Version.Release GE 5.3 THEN BEGIN ENDIF

However, a colleague tells me: no chance! The program will report a syntax error when run in

IDL 5.2! So, the problem is that IDL looks for a function of this name during compile stage

and doesn't care if it will ever be used. Only help is to write two little dummy functions

named stregex.pro and strmatch.pro which do nothing but print a warning message. And so I

did. Works nicely, and you don't even have to test for the release number because these two

functions are internal in 5.3 and hence they are discovered before anything on disk can be found.

Just a little bedtime story ;-)

Cheers, Martin

```
[[ Dr. Martin Schultz Max-Planck-Institut fuer Meteorologie
             Bundesstr. 55, 20146 Hamburg
             phone: +49 40 41173-308
[]
                                               \prod
             fax: +49 40 41173-298
                                              [[
\prod
[[ martin.schultz@dkrz.de
                                              []
Subject: Re: David: for your book?
Posted by Craig Markwardt on Tue, 15 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message
Martin Schultz <martin.schultz@dkrz.de> writes:
> Hi,
>
    Testing IDL's capabilities is always dangerous: you might get more
 than you ask for!
>
> What would you expect to get out of the following:
> test = [ 'dir1/', 'directory2/', 'another_directory/' ]
> print,strmid(test,0,strlen(test)-1)
Hmm, I got an error message!
IDL> test = [ 'dir1/', 'directory2/', 'another_directory/' ]
IDL> print,strmid(test,0,strlen(test)-1)
% STRMID: Expression must be a scalar in this context: <LONG
                                                              Array[3]>.
% Execution halted at: $MAIN$
IDL> print, !version
{ alpha OSF unix 5.2 Oct 30 1998}
Is this something new in IDL 5.3?
Craig
```

Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu

Subject: Re: David: for your book?
Posted by landsman on Wed, 16 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message

In article <399937D2.8C0A52A2@dkrz.de>.

```
Martin Schultz <martin.schultz@dkrz.de> wrote:
>
> What would you expect to get out of the following:
>
> test = [ 'dir1/', 'directory2/', 'another_directory/' ]
> print,strmid(test,0,strlen(test)-1)
>
> ??
```

- > Well, I at least had expected to get the directory names without the
- > trailing '/'.
- > But, no!
- > help, strmid(test,0,strlen(test)-1)
- > reveals that the result is a 3x3 array!!

When IDL added vector capabilities to STRMID in V5.3, they allowed for both (1) the extraction of multiple substrings from each element of a string, and (2) having the first character position and substring length differ for each element of a string array. Unfortunately, adding all this power has made the syntax awkward. According to the documentation

"If First_Character or Length is an array, the size of their first dimension determines how many substrings are extracted from each element of <expression>"

So the solution is to REFORM the string length (or position) array to two dimensions:

```
IDL> test = [ 'dir1/', 'directory2/', 'another_directory/']
IDL> print,strmid(test,0,reform( strlen(test)-1, 1, 3) )
===>dir1 directory2 another_directory
```

By reforming the string length array into a 1 x 3 size, STRMID now knows to extract only 1 substring from each element of test.

A simple program to extract a string array up to a specified character using the vector capabilities of STRMID is available at

http://idlastro.gsfc.nasa.gov/ftp/v53/gettok.pro

--Wayne Landsman landsman@mpb.gsfc.nasa.gov

Sent via Deja.com http://www.deja.com/

Subject: Re: David: for your book?

Posted by davidf on Wed, 16 Aug 2000 07:00:00 GMT

View Forum Message <> Reply to Message

Martin Schultz (martin.schultz@dkrz.de) writes:

- > BTW: any one out there who is still using earlier versions (prior to
- > 5.2)?

Jeeze Louise, man! Have you checked with your systems administrator? You are about to have a denial of service attack from all the e-mail about to pile up in your account! :-)

Cheers.

David

--

David Fanning, Ph.D.

Fanning Software Consulting

Phone: 970-221-0438 E-Mail: davidf@dfanning.com

Coyote's Guide to IDL Programming: http://www.dfanning.com/

Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: David: for your book?

Posted by Martin Schultz on Wed, 16 Aug 2000 07:00:00 GMT

View Forum Message <> Reply to Message

Craig Markwardt wrote:

>

- > Martin Schultz <martin.schultz@dkrz.de> writes:
- >> As an aside, I recently found another problem in terms of backward
- >> compability (I am
- >> sure something similar was already mentioned in this newsgroup, but this
- >> was the first
- >> time I had this problem):
- >> I tried to write a routine that takes advantage of the new StrMatch
- >> and StRegex
- >> functions in 5.3. In order to ensure backward compatibility, I enclosed
- >> these function calls
- >> with
- >> IF !Version.Release GE 5.3 THEN BEGIN
- >> ENDIF

>> >> However, a colleague tells me: no chance! The program will report a >> syntax error when run in >> IDL 5.2! So, the problem is that IDL looks for a function of this name >> during compile stage >> and doesn't care if it will ever be used. > > > There is a way to do this without defining an external function, by > using FORWARD FUNCTION properly: > > FORWARD FUNCTION stregex, strmatch > IF !Version.Release GE 5.3 THEN BEGIN

- foo = stregex(bar)
- sm = strmatch(eg)
- > ENDIF

- > This will make sure that IDL will always recognize strmatch & stregex
- > as functions during the compilation stage, and there will be no syntax
- > errors in any IDL version. It is harmless in IDL 5.3. If you really
- > want the warning message, then by all means go ahead and make your own
- > strmatch definitions et al.

- > Have fun.
- > Craig

Thanks, Craig.

I didn't believe that this would cure it (I had thought it would just change the

error message), but you are right, it works. So, maybe some people can still live

with IDL 5.2 after all.

BTW: any one out there who is still using earlier versions (prior to 5.2)?

Maybe I shouldn't ask because I fear this will swamp the newsgroup -- so please

reply only by email, subject "old idl", content '!version = ' then I will

collect these messages for a while and report a summary later (after a week

or two). I assure you that these messages will be kept strictly confidental and

will be deleted after processing the version information.

Cheers. Martin

Page 8 of 10 ---- Generated from comp.lang.idl-pvwave archive

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

Subject: Re: David: for your book? Posted by R.Bauer on Wed, 16 Aug 2000 07:00:00 GMT View Forum Message <> Reply to Message

```
Craig Markwardt wrote:
> Martin Schultz <martin.schultz@dkrz.de> writes:
>> As an aside, I recently found another problem in terms of backward
>> compability (I am
>> sure something similar was already mentioned in this newsgroup, but this
>> was the first
>> time I had this problem):
     I tried to write a routine that takes advantage of the new StrMatch
>> and StRegex
>> functions in 5.3. In order to ensure backward compatibility, I enclosed
>> these function calls
>> with
      IF !Version.Release GE 5.3 THEN BEGIN
>>
      ENDIF
>>
>>
>> However, a colleague tells me: no chance! The program will report a
>> syntax error when run in
>> IDL 5.2! So, the problem is that IDL looks for a function of this name
>> during compile stage
>> and doesn't care if it will ever be used.
> ...
>
> There is a way to do this without defining an external function, by
> using FORWARD_FUNCTION properly:
> FORWARD_FUNCTION stregex, strmatch
> IF !Version.Release GE 5.3 THEN BEGIN
    foo = stregex(bar)
    sm = strmatch(eg)
```

ENDIF
This will make sure that IDL will always recognize strmatch & stregex
as functions during the compilation stage, and there will be no syntax
errors in any IDL version. It is harmless in IDL 5.3. If you really
want the warning message, then by all means go ahead and make your own
strmatch definitions et al.
Have fun,
Craig

Don't write it yourself

Take the strmatch routine from the deutsch library http://www.astro.washington.edu/deutsch/idl/htmlhelp/slibrar y24.html

Reimar