Subject: forcing variable definition in IDL?
Posted by Gernot Hassenpflug on Tue, 01 May 2001 06:19:42 GMT
View Forum Message <> Reply to Message

I have seen some threads on Google about the lack of 'implicit none'
Fortran type statement in IDL. However, those threads are, a) short,
b) date from 1995. I have not seen anything later than this.

So, my question: is it possible in IDL 5.4 to force definition of
variables, or at least to automate a variable-check in IDL.

Alternatively, maybe I need to write a function that automatically
checks the first use of a variable in a program. I have used the
routine_info and routine_names functions to obtain information
about variables in scope at the time.

Does anyone know of either how to get IDL to check variables'
definition,
or how to write a function to do that?

I have also toyed with the idea of using each variable as a single-
element array (eg. b(0) = 1), but that is most inelegant and lengthy.

Best regards,
Gernot Hassenpflug
PS email address in header includes SPAM TRAP. please note.
--
Gernot Hassenpflug, MSc. (Eng.)
+81 (0)75 213-1321 (H) Tel/Fax
+81 (0)774 38-3868 (W) Tel
gernot@kurasc.kyoto-u.ac.jp
Radio Atmospheric Science Centre, Kyoto University

Subject: Re: forcing variable definition in IDL?
Posted by davidf on Tue, 01 May 2001 12:32:13 GMT
View Forum Message <> Reply to Message

Gernot Hassenpflug (gernot-nospam@kurasc.kyoto-u.ac.jp) writes:

> So, my question: is it possible in IDL 5.4 to force definition of
> variables, or at least to automate a variable-check in IDL.

It is certainly possible to "force" a variable to
have a certain definition. That's not the problem.
The problem is making it *stay* a certain definition.
That, in general, is not possible in IDL, since IDL

has dynamic typing capability.
>
> Alternatively, maybe I need to write a function that automatically
> checks the first use of a variable in a program. I have used the
> routine_info and routine_names functions to obtain information
> about variables in scope at the time.

You can check. But like recalcitrant children, they
will be doing something else the minute you turn
your back. :-)

> Does anyone know of either how to get IDL to check variables'
> definition,
> or how to write a function to do that?

```
IDL> theType = Size(variable, /Type)
```

Or, if you prefer the type "name":

```
IDL> theTypeName = Size(variable, /TName)
```

> I have also toyed with the idea of using each variable as a single-
> element array (eg. b(0) = 1), but that is most inelegant and lengthy.

Don't bother. IDL scalars *are* single element arrays:

```
IDL> a=5
IDL> a[0] = 6 & Print, a
```

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: forcing variable definition in IDL?
Posted by William Daffer on Tue, 01 May 2001 17:18:10 GMT
View Forum Message <> Reply to Message

davidf@dfanning.com (David Fanning) writes:
[...]

> Don't bother. IDL scalars *are* single element arrays:

>
>   IDL> a=5
>   IDL> a[0] = 6 & Print, a
>

  Um... Not true.

IDL> a=['foo|bar']
IDL> print,strsplit(a,'|',/extract)
% STRTOK: Expression must be a scalar in this context: STRING.
% Execution halted at: STRSPLIT   24
  /usr/local/rsi/idl_5.3/lib/strsplit.pro
% $MAIN$
IDL> retall
IDL> print,strsplit(a[0],'|',/extract)
foo bar
IDL>

  There are some other RSI supplied code where one sees this behavior.

  By the way, this is idl 5.3. I haven't checked idl 5.4.

whd
--
Outside of a dog a book is man's best friend.
Inside of a dog it's too dark to read
                        Groucho Marx

Subject: Re: forcing variable definition in IDL?
Posted by Liam E. Gumley on Tue, 01 May 2001 18:19:53 GMT
View Forum Message <> Reply to Message

William Daffer wrote:
>
> davidf@dfanning.com (David Fanning) writes:
> [...]
>
>> Don't bother. IDL scalars *are* single element arrays:
>>
>>   IDL> a=5
>>   IDL> a[0] = 6 & Print, a
>>
>
>   Um... Not true.
>
> IDL> a=['foo|bar']
> IDL> print,strsplit(a,'|',/extract)

> % STRTOK: Expression must be a scalar in this context: STRING.
> % Execution halted at: STRSPLIT 24
> /usr/local/rsi/idl_5.3/lib/strsplit.pro
> % $MAIN$
> IDL> retall
> IDL> print,strsplit(a[0],'|',/extract)
> foo bar
> IDL>
>
> There are some other RSI supplied code where one sees this behavior.
>
> By the way, this is idl 5.3. I haven't checked idl 5.4.

An array with one element is an *array*, i.e., it has one dimension:

```
IDL> a = [25]
IDL> help, a
A               INT       = Array[1]
IDL> print, size(a, /n_dimensions)
          1
```

A single subscripted array element is a *scalar expression*, i.e., it
has no dimensions:

```
IDL> a = [1, 2, 3, 4, 5]
IDL> help, a[0]
<Expression>    INT       =        1
IDL> print, size(a[0], /n_dimensions)
          0
```

A scalar may be treated as though it were a single subscripted array
element. However, as shown above, a scalar expression has no dimensions:

```
IDL> a = 100
IDL> help, a
A               INT       =      100
IDL> help, a[0]
<Expression>    INT       =      100
IDL> print, size(a, /n_dimensions)
          0
IDL> print, size(a[0], /n_dimensions)
          0
```

The implementer of STRTOK (which is called by STRSPLIT) is therefore
checking for an input argument which has no dimensions.

Cheers,
Liam.

Subject: Re: forcing variable definition in IDL?
Posted by William Daffer on Tue, 01 May 2001 20:26:29 GMT
View Forum Message <> Reply to Message

"Liam E. Gumley" <Liam.Gumley@ssec.wisc.edu> writes:

> William Daffer wrote:
>>
>> davidf@dfanning.com (David Fanning) writes:
>> [...]
>>
>>> Don't bother. IDL scalars *are* single element arrays:
>>>
>>> IDL> a=5
>>> IDL> a[0] = 6 & Print, a
>>>
>>
>> Um... Not true.
>>
>> IDL> a=['foo|bar']
>> IDL> print,strsplit(a,'|',/extract)
>> % STRTOK: Expression must be a scalar in this context: STRING.
>> % Execution halted at: STRSPLIT 24
>> /usr/local/rsi/idl_5.3/lib/strsplit.pro
>> % $MAIN$
>> IDL> retall
>> IDL> print,strsplit(a[0],'|',/extract)
>> foo bar
>> IDL>
>>
>> There are some other RSI supplied code where one sees this behavior.
>>
>> By the way, this is idl 5.3. I haven't checked idl 5.4.
>
> An array with one element is an *array*, i.e., it has one dimension:
>
> IDL> a = [25]
> IDL> help, a
> A INT = Array[1]
> IDL> print, size(a, /n_dimensions)
> 1
>
> A single subscripted array element is a *scalar expression*, i.e., it
> has no dimensions:
>

> IDL> a = [1, 2, 3, 4, 5]
> IDL> help, a[0]
> <Expression> INT = 1
> IDL> print, size(a[0], /n_dimensions)
> 0
>
> A scalar may be treated as though it were a single subscripted array
> element. However, as shown above, a scalar expression has no dimensions:
>
> IDL> a = 100
> IDL> help, a
> A INT = 100
> IDL> help, a[0]
> <Expression> INT = 100
> IDL> print, size(a, /n_dimensions)
> 0
> IDL> print, size(a[0], /n_dimensions)
> 0
>
> The implementer of STRTOK (which is called by STRSPLIT) is therefore
> checking for an input argument which has no dimensions.
>
> Cheers,
> Liam.
> http://cimss.ssec.wisc.edu/~gumley/

Um... so you're agreeing with me when I say that David's remark is untrue?

whd
--
Outside of a dog a book is man's best friend.
Inside of a dog it's too dark to read
Groucho Marx

Subject: Re: forcing variable definition in IDL?
Posted by davidf on Tue, 01 May 2001 20:40:35 GMT
View Forum Message <> Reply to Message

William Daffer (whdaffer@mediaone.net) writes:

> Um... so you're agreeing with me when I say that David's remark is
> untrue?

I think Liam was trying to say that my remark was only
half true, in a smart-alacky sort of way. And that your

comment pointed that out. :-)

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
