Subject: Re: arrays vs. functions conflicts
Posted by David Fanning on Fri, 05 Mar 2004 13:56:30 GMT
View Forum Message <> Reply to Message

Paolo Grigis writes:

> Thus my problem:
> To resolve the conflict {that is, limits.pro being already
> compiled and procedure.pro refusing to compile because it has
> an old-fashioned statement like var=limits(1:2) instead of
> var=limits[1:2]} I'm thinking of automatically compiling
> the (hopefully) few troubling routines like procedure.pro
> at startup using the resolve_routine() statement.
> (BTW, why is the IDL compiler (5.5) not smart enough to
> understand that function(1:2) is an array? ":" is never allowed
> in function calls, after all.)
>
> But before going on, I just wanted to know if there is an
> easier way ouy of this that I have overlooked.

IDL itself could care less about this issue. So if
you are having problems with it then *you* must
care about it. Does you procedure.pro have a compiler
option that forces strict arrays? Then take it out.
Problem solved. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: arrays vs. functions conflicts
Posted by Paolo Grigis on Fri, 05 Mar 2004 14:22:07 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Paolo Grigis writes:
>
>
>> Thus my problem:
>> To resolve the conflict {that is, limits.pro being already
>> compiled and procedure.pro refusing to compile because it has
>> an old-fashioned statement like var=limits(1:2) instead of
>> var=limits[1:2]} I'm thinking of automatically compiling

>> the (hopefully) few troubling routines like procedure.pro
>> at startup using the resolve_routine() statement.
>> (BTW, why is the IDL compiler (5.5) not smart enough to
>> understand that function(1:2) is an array? ":" is never allowed
>> in function calls, after all.)
>>
>> But before going on, I just wanted to know if there is an
>> easier way ouy of this that I have overlooked.
>
>
> IDL itself could care less about this issue. So if
> you are having problems with it then *you* must
> care about it. Does you procedure.pro have a compiler
> option that forces strict arrays? Then take it out.
> Problem solved. :-)
>
> Cheers,
>
> David

No, the routine compiles just fine if there aren't
any previously compiled functions called "limits", but
fails to compile if this is the case, because then IDL
thinks it is a function instead of an array.
Hence I was thinking of compiling the routine at
the idl start: if I do that then I don't have any problems
at all.

Cheers,
Paolo

--

Paolo Grigis
ETHZ - Institute of Astronomy email: pgrigis@astro.phys.ethz.ch
Scheuchzerstrasse 7
ETH Zentrum phone: ++41 1 632 42 20
8092 Zurich fax : ++41 1 632 12 05
Switzerland http://www.astro.phys.ethz.ch/

Subject: Re: arrays vs. functions conflicts

Posted by David Fanning on Fri, 05 Mar 2004 14:37:15 GMT
View Forum Message <> Reply to Message

Paolo Grigis writes:

> No, the routine compiles just fine if there aren't
> any previously compiled functions called "limits", but
> *fails* to compile if this is the case, because then IDL
> thinks it is a function instead of an array.

I'm saying it thinks so because you told it to think so.
IDL is not this bright on its own. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: arrays vs. functions conflicts
Posted by David Fanning on Fri, 05 Mar 2004 14:39:34 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> I'm saying it thinks so because you told it to think so.
> IDL is not this bright on its own. :-)

"Bright" is not the right word. IDL is not this
discriminating on its own.

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: arrays vs. functions conflicts
Posted by Paolo Grigis on Fri, 05 Mar 2004 14:54:35 GMT
View Forum Message <> Reply to Message

David Fanning wrote:

> David Fanning writes:
>
>
>> I'm saying it thinks so because you told it to think so.
>> IDL is not this bright on its own. :-)
>
>
> "Bright" is not the right word. IDL is not this
> *discriminating* on its own.
>
> Cheers,
>
> David

Allright, I know this is difficult for IDL, but I would
just like to know if there is a way out of the problem...
Ok, maybe I was just being too optimistic, and there is
not a solution beside hoping that everybody will switch to
the []-bracket notation as soon as possible. In the
meantime I suppose I have to be careful on the order
of compilation of the routines.

In any case, there aren't any compiler directives in the
main procedure.

Cheers,
Paolo

Subject: Re: arrays vs. functions conflicts
Posted by Paul Van Delst[1] on Fri, 05 Mar 2004 14:54:58 GMT
View Forum Message <> Reply to Message

Paolo Grigis wrote:
>
> David Fanning wrote:
>> Paolo Grigis writes:
>>
>>
>>> Thus my problem:
>>> To resolve the conflict {that is, limits.pro being already
>>> compiled and procedure.pro refusing to compile because it has
>>> an old-fashioned statement like var=limits(1:2) instead of
>>> var=limits[1:2]} I'm thinking of automatically compiling
>>> the (hopefully) few troubling routines like procedure.pro
>>> at startup using the resolve_routine() statement.
>>> (BTW, why is the IDL compiler (5.5) not smart enough to
>>> understand that function(1:2) is an array? ":" is never allowed

>>> in function calls, after all.)
>>>
>>> But before going on, I just wanted to know if there is an
>>> easier way ouy of this that I have overlooked.
>>
>>
>> IDL itself could care less about this issue. So if
>> you are having problems with it then *you* must
>> care about it. Does you procedure.pro have a compiler
>> option that forces strict arrays? Then take it out.
>> Problem solved. :-)
>>
>> Cheers,
>>
>> David
>
> No, the routine compiles just fine if there aren't
> any previously compiled functions called "limits", but
> *fails* to compile if this is the case, because then IDL
> thinks it is a function instead of an array.
> Hence I was thinking of compiling the routine at
> the idl start: if I do that then I don't have any problems
> at all.

Hello Paolo,

Is there a COMPILE STRICTARR directive anywhere in your code or in any startup scripts? This is the most obvious source of weirdness between [] and (). But.....

Hang on a minute.... you say you have a limits.pro that compiles. Thus "limits" _is_ a function, right? The you have a statement like var=limits(1:2) where "limits" is now an array? Well, which do you want "limits" to be...a function or an array?

Confusedly yours,

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC

Subject: Re: arrays vs. functions conflicts
Posted by David Fanning on Fri, 05 Mar 2004 15:04:18 GMT
View Forum Message <> Reply to Message

Paolo Grigis writes:

> Allright, I know this is difficult for IDL, but I would
> just like to know if there is a way out of the problem...
> Ok, maybe I was just being too optimistic, and there is
> not a solution beside hoping that everybody will switch to
> the []-bracket notation as soon as possible. In the
> meantime I suppose I have to be careful on the order
> of compilation of the routines.
>
> In any case, there aren't any compiler directives in the
> main procedure.

Well, somewhere then. IDL just doesn't figure this out
on its own. Are you sure the problem has to do with
square brackets and not naming conventions? Is the offending
program in its own file, with the file having the same
name as the program module?

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: arrays vs. functions conflicts
Posted by Paolo Grigis on Fri, 05 Mar 2004 15:17:14 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:
> Paolo Grigis wrote:
>
>> David Fanning wrote:
>>
>>> Paolo Grigis writes:
>>>
>>>
>>>
>>>> Thus my problem:
>>>> To resolve the conflict {that is, limits.pro being already
>>>> compiled and procedure.pro refusing to compile because it has
>>>> an old-fashioned statement like var=limits(1:2) instead of
>>>> var=limits[1:2]} I'm thinking of automatically compiling
>>>> the (hopefully) few troubling routines like procedure.pro
>>>> at startup using the resolve_routine() statement.
>>>> (BTW, why is the IDL compiler (5.5) not smart enough to
>>>> understand that function(1:2) is an array? ":" is never allowed

>>>> in function calls, after all.)
>>>>
>>>> But before going on, I just wanted to know if there is an
>>>> easier way ouy of this that I have overlooked.
>>>
>>>
>>> IDL itself could care less about this issue. So if
>>> you are having problems with it then *you* must
>>> care about it. Does you procedure.pro have a compiler
>>> option that forces strict arrays? Then take it out.
>>> Problem solved. :-)
>>>
>>> Cheers,
>>>
>>> David
>>
>> No, the routine compiles just fine if there aren't
>> any previously compiled functions called "limits", but
>> *fails* to compile if this is the case, because then IDL
>> thinks it is a function instead of an array.
>> Hence I was thinking of compiling the routine at
>> the idl start: if I do that then I don't have any problems
>> at all.
>
>
> Hello Paolo,
>
> Is there a COMPILE STRICTARR directive anywhere in your code or in any startup scripts?
> This is the most obvious source of weirdness between [] and (). But.....
>
> Hang on a minute.... you say you have a limits.pro that compiles. Thus "limits" _is_ a
> function, right? The you have a statement like var=limits(1:2) where "limits" is now an
> array? Well, which do you want "limits" to be...a function or an array?
>
> Confusedly yours,
>
> paulv
>

Dear Paul, that's the problem: in the procedure, limits is an
array. No problem with that, it should be.

But sometimes it happens that I run *another* program before,
which compile a function called limits. If then I compile the
procedure, the compiler thinks limits is a function and all
hell break loose.

The problem is that in general you don't have control on what

other people define as a function, and so this problem does
happen. The only way out I see, is to exit idl, and compile
the procedure before the function.

Cheers,
Paolo

--

Paolo Grigis
ETHZ - Institute of Astronomy email: pgrigis@astro.phys.ethz.ch
Scheuchzerstrasse 7
ETH Zentrum phone: ++41 1 632 42 20
8092 Zurich fax : ++41 1 632 12 05
Switzerland http://www.astro.phys.ethz.ch/

Subject: Re: arrays vs. functions conflicts
Posted by Paolo Grigis on Fri, 05 Mar 2004 15:26:34 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Paolo Grigis writes:
>
>
>> Allright, I know this is difficult for IDL, but I would
>> just like to know if there is a way out of the problem...
>> Ok, maybe I was just being too optimistic, and there is
>> not a solution beside hoping that everybody will switch to
>> the []-bracket notation as soon as possible. In the
>> meantime I suppose I have to be careful on the order
>> of compilation of the routines.
>>
>> In any case, there aren't any compiler directives in the
>> main procedure.
>
>
> Well, somewhere then. IDL just doesn't figure this out
> on its own. Are you sure the problem has to do with
> square brackets and not naming conventions? Is the offending
> program in its own file, with the file having the same
> name as the program module?
>
> Cheers,
>
> David

Dear David,
I made a simple example which shows the problem:

```
--------------------------------
;file test1.pro
FUNCTION test1

return,-1

END

--------------------------------
;file test2.pro
PRO test2

test1=[5,4,6]

print,test1(0:1)

END
--------------------------------
```

After strting idl:

```
;this fails
print,test1()
test2

;output:
IDL> print,test1()
% Compiled module: TEST1.
      -1
IDL> test2

print,test1(0:1)
             ^
% Syntax error.
   At: /users/pgrigis/test2.pro, Line 5
% Compiled module: TEST2.
% Attempt to call undefined procedure/function: 'TEST2'.
% Execution halted at:  $MAIN$
```

compare with:

```
;this works
```

```
test2
print,test1()
test2


;output
IDL> test2
% Compiled module: TEST2.
       5       4
IDL> print,test1()
% Compiled module: TEST1.
      -1
IDL> test2
       5       4
```

Quite bad, isn't it?

Cheers,
Paolo

--

Paolo Grigis
ETHZ - Institute of Astronomy email: pgrigis@astro.phys.ethz.ch
Scheuchzerstrasse 7
ETH Zentrum phone: ++41 1 632 42 20
8092 Zurich fax : ++41 1 632 12 05
Switzerland http://www.astro.phys.ethz.ch/

Subject: Re: arrays vs. functions conflicts
Posted by David Fanning on Fri, 05 Mar 2004 15:44:41 GMT
View Forum Message <> Reply to Message

Paolo Grigis writes:

> But sometimes it happens that I run *another* program before,
> which compile a function called limits. If then I compile the
> procedure, the compiler thinks limits is a function and all
> hell break loose.

Ah, yes, well. This is a problem not so much

with square brackets (which obviously help eliminate
this problem), but with the fact that IDL is perfectly
happy with arrays and functions (and procedures, for
that matter) having the same name.

Once a name is put into IDL's memory space as an array,
it is going to be impossible to get the same name registered
in its memory space as a function unless you explicitly
compile the function. (Exiting IDL is not necessary.)

Then, of course, since it is listed in IDL as both a
variable AND a function, you are going to have trouble
telling one from the other without using square brackets.

We have been trying to solve the second problem first. :-)

The first problem can be solved by using a FORWARD_FUNCTION
statement. And, if you are lucky, unlike some people we know,
that might also solve your second problem.

And, of course, you can also write a polite e-mail to the
people supplying your code (Craig!!) to please get with the
21st century and put square brackets on their arrays. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: arrays vs. functions conflicts
Posted by Wayne Landsman on Fri, 05 Mar 2004 15:47:56 GMT
View Forum Message <> Reply to Message

> But before going on, I just wanted to know if there is an
> easier way ouy of this that I have overlooked (and no, I
> definitely do not want to override useful programs written
> by others with my own []-version, only to start all over
> again each time a new version or bug fix of the routine
> comes out).

You might try the program idlv4_to_v5.pro in
http://idlastro.gsfc.nasa.gov/ftp/contrib/landsman/v5/
which does an automatic replacement of () subscripts with [] using

ROUTINE_INFO() to check whether a variable is an array or a function.
It works about 99% of the time though occasionally gets confused by
multiple lines or execute statements. You can then add a
compile_opt idl2 to ensure that IDL knows the difference.
You might then give the [] version to the developers of the routines and
tell them that "nobody is using V4.0 or earlier anymore...."

--Wayne

Subject: Re: arrays vs. functions conflicts
Posted by David Fanning on Fri, 05 Mar 2004 15:53:32 GMT
View Forum Message <> Reply to Message

Paolo Grigis writes:

> Quite bad, isn't it?

Well, it comes up occasionally, but it is always easily solved
by a global search and replace on a variable name.

The alternative to have have variable naming restrictions, so
that variables could only be given names that hadn't already
been used for something else. That would put us in the same
mess that trendy parents have in naming their children. :-(

Cheers,

David

P.S. Let's just say I'm glad I don't have to go through
life with a misspelled name just to be unique.

--
David Fanning, Ph.D.
Fanning Software Consulting
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: arrays vs. functions conflicts
Posted by Paolo Grigis on Fri, 05 Mar 2004 16:38:02 GMT
View Forum Message <> Reply to Message

Wayne Landsman wrote:
>
>> But before going on, I just wanted to know if there is an
>> easier way ouy of this that I have overlooked (and no, I

>> definitely do not want to override useful programs written
>> by others with my own []-version, only to start all over
>> again each time a new version or bug fix of the routine
>> comes out).
>
>
> You might try the program idlv4_to_v5.pro in
> http://idlastro.gsfc.nasa.gov/ftp/contrib/landsman/v5/
> which does an automatic replacement of () subscripts with [] using
> ROUTINE_INFO() to check whether a variable is an array or a function.
> It works about 99% of the time though occasionally gets confused by
> multiple lines or execute statements. You can then add a
> compile_opt idl2 to ensure that IDL knows the difference.
> You might then give the [] version to the developers of the routines and
> tell them that "nobody is using V4.0 or earlier anymore...."
>
> --Wayne
>

Dear David, Wayne, Paul

thanks for your replies.

Well, that's exactly what I feared: no way out!

Just to show you how common the problem is one can try:

utplot,[1,2,3],[1,2,3],0
mpfit

or, if you don't have utplot,

.comp
function limits
end

mpfit

Even if utplot might not be as widespread as mpfit,I guess there are
quite a few people around using both of them...

Dear Craig,

should you ever feel like giving up idl 4, I think you could be
surprised how many people might volunteer to help you translating
some code to []-notation!

Greetings,
Paolo

--

Paolo Grigis
ETHZ - Institute of Astronomy email: pgrigis@astro.phys.ethz.ch
Scheuchzerstrasse 7
ETH Zentrum phone: ++41 1 632 42 20
8092 Zurich fax : ++41 1 632 12 05
Switzerland http://www.astro.phys.ethz.ch/

Subject: Re: arrays vs. functions conflicts
Posted by Bruce Bowler on Fri, 05 Mar 2004 19:24:27 GMT
View Forum Message <> Reply to Message

pet peeve alert...

On Fri, 05 Mar 2004 06:56:30 -0700, David Fanning put fingers to keyboard
and said:

> IDL itself could care less about this issue.

IDL could *NOT* care less about. The way you wrote it IDL, since it could
care less, must care some, and we know IDL doesn't give a rats arse...

We now return you to your regularly scheduled discussion.

end pet peeve alert

--

+------------------+------------------------------------ ------------+
Bruce Bowler | If you're walking on eggs, don't hop - Anonymous
1.207.633.9600 |
bbowler@bigelow.org |
+------------------+------------------------------------ ------------+
