Subject: Am I stupid?
Posted by colinr on Tue, 10 Jul 2001 13:29:53 GMT
View Forum Message <> Reply to Message

pro stupid,a=a,a1=a1

print,'a=',a
print,'a1=',a1

end

IDL> stupid,a=1,a1=2
% Ambiguous keyword abbreviation: A.

so is there a sensible way around this? What I'm actually trying to
do is to have a keyword AXISCOLOR in a routine that also accepts
the AX keyword. I know I could call it something different but am I
just being a whiny luser if I think the above code should work
as is?

--
Colin Rosenthal
Astrophysics Institute
University of Oslo

Subject: Re: Am I stupid?
Posted by gorkom on Wed, 11 Jul 2001 21:04:22 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:
> If I understand you (and Jaco) correctly, what you're saying is:
>
> The abbreviated keyword is too short to be unique, therefore it is.

That just about sums it up, almost. Let me rephrase it to:

The keyword is too short to be unique, therefore it is unabbreviated.

> I think this
> is a case where statements can be neither proved nor disproved.... :o)
In this case, the IDL interpreter proves you right. Whereas usually you are right.
Jaco

Subject: Re: Am I stupid?

Posted by James Kuyper on Wed, 11 Jul 2001 22:04:24 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:
>
> James Kuyper wrote:
...
>> It would know that TIME cannot be an abbreviated keyword, because it's
>> too short to be unique. Therefore, it must be the full keyword.
>
> If I understand you (and Jaco) correctly, what you're saying is:
>
> The abbreviated keyword is too short to be unique, therefore it is.

Incorrect. The correct wording would be:

The keyword is too short to be a unique abbreviation, therefore it isn't
an abbreviation.

Subject: Re: Am I stupid?
Posted by William Clodius on Wed, 11 Jul 2001 22:14:01 GMT
View Forum Message <> Reply to Message

A better way to describe what could, but is not, implemented for IDL's keywords is a multise stage process

1. If a keyword on an invocation exactly matches a keyword in the functions's definition then it is that keyword, else

2. if the keyord on invocation is an abbreviation for exactly one keyword in the functions's definitition then it is that keyword, else

3. It is an error that can be determined statically.

Subject: Re: Am I stupid?
Posted by William Clodius on Wed, 11 Jul 2001 22:23:26 GMT
View Forum Message <> Reply to Message

William Thompson wrote:

> Bob Crawford <r_w_crawford@yahoo.com> writes:
>
>> H C Pumphrey wrote:
>
>>> In article <onn16cvpe1.fsf@cow.physics.wisc.edu>, Craig Markwardt
<craigmnet@cow.physics.wisc.edu> writes:

>>> |> Not that [IDL's policy on keyword uniqueness] doesn't frustrate the heck
>>> |> out of me sometimes. There
>>> |> are many times where I want keywords like TIME, TIMEUNIT, TIMESTEP,
>>> |> and so on. My suggestion is that the above policy should hold,
>>> |> *unless* there is an exact match to a specific keyword.
>>>
>>> That seems to be what R [1] does already. Apart from this special case, I don't
>>> see any way around it, not without IDL having a direct link to your
>>> subconscious so that it can tell which out of TIME, TIMEUNIT and TIMESTEP
>>> you meant when you said FOO,TI=137
>
>> ... but it should know what you meant when you say: FOO, TIME=37
>
>> Bob.
>> (sometimes frustrated too)
>
> It is quite possible to redefine the rules such that one could unambiguously
> have the keywords TIME and TIMESTEP simultaneously. However, consider what
> happens when a user mistakenly abbreviates TIMESTEP=3 to TIME=3. Under the
> present rules, IDL signals that a possible error has occured. But with the
> proposed new rules, no error message would be generated--it would just happily
> go ahead and do the wrong thing!

<snip>

However the current rules are also error prone. Single character typos are common, and the allowal of
abbreviations means that arguments that disagree in more than one character, can still be confused with one
another by a single character mistake. In particular I am prone to accidently converting singulars to plurals
and vice versa. In this case, I suspect that I would frequently type TIMES=37 and have it interpretted as
TIMESTEP when I meant TIME.

Subject: Re: Am I stupid?
Posted by david[2] on Wed, 11 Jul 2001 22:48:47 GMT
View Forum Message <> Reply to Message

William Clodius writes:

> However the current rules are also error prone. Single character typos are common, and the allowal of
> abbreviations means that arguments that disagree in more than one character, can still be confused with one
> another by a single character mistake. In particular I am prone to accidently converting singulars to plurals

> and vice versa. In this case, I suspect that I would frequently type TIMES=37 and have it interpretted as
> TIMESTEP when I meant TIME.

I would be *extremely* happy if absolutely NO
keyword abbreviations were allowed. At least I
would be able to read and understand code that
I was looking at. :-)

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: Am I stupid?
Posted by m.hadfield on Thu, 12 Jul 2001 02:20:13 GMT
View Forum Message <> Reply to Message

From: "David Fanning" <david@dfanning.com>
>
> I would be *extremely* happy if absolutely NO
> keyword abbreviations were allowed.

You must admit they are handy at the command line.

> At least I
> would be able to read and understand code that
> I was looking at. :-)

"Doctor, will I play the piano when the cast comes off my hand?"
"Certainly."
"Oh good, I never could play it before."

Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield
National Institute for Water and Atmospheric Research

--
Posted from clam.niwa.cri.nz [202.36.29.1]
via Mailgate.ORG Server - http://www.Mailgate.ORG

Subject: Re: Am I stupid?
Posted by Craig Markwardt on Thu, 12 Jul 2001 05:06:12 GMT
View Forum Message <> Reply to Message

david@dfanning.com (David Fanning) writes:
> I would be *extremely* happy if absolutely NO
> keyword abbreviations were allowed. At least I
> would be able to read and understand code that
> I was looking at. :-)

Unfortunately, even a person who uses no abbreviations may be caught
by the current policy. Which is to say, even if I faithfully used
the TIME, TIMESTEP and TIMEUNIT keywords unabbreviated, I would still
be in a hurtful world. :-)

If the policy is really that TIME is ambiguous, then it shouldn't be
allowed to be compiled in the first place.

Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------

Subject: Re: Am I stupid?
Posted by colinr on Thu, 12 Jul 2001 07:23:40 GMT
View Forum Message <> Reply to Message

On 11 Jul 2001 21:02:04 GMT,
William Thompson <thompson@orpheus.nascom.nasa.gov> wrote:

> I've also been occasionally frustrated by this behavior, but I've come to the
> conclusion (a little reluctantly) that it's actually ``A Good Thing''. In a
> perfect world, we could be sure that the user remembered not to use TIME as an
> abbreviation of TIMESTEP, but you and I know that we don't live in a perfect
> world.
>
> It would be better, as somebody has already suggested, if situations like this
> were caught by the compiler, instead of waiting until somebody actually tried

> to use the TIME keyword. But the present abbreviation rules should not be
> changed.

However the current situation also allows the related problem I
described:

pro MySurfacePlot,axiscolor=axiscolor,_extra=e

<do some stuff with colors>
shade_surf,MyData,_extra=e

end

which works perfectly for the first 100 times the user uses it until
the day he trys to set the AX keyword. Thus to use _extra correctly one
must bear in mind all possible valid keywords to all inheriting
subroutines and ensure that none of programmer-defined keywords
can be abbreviated as any of them.

--
Colin Rosenthal
Astrophysics Institute
University of Oslo

Subject: Re: Am I stupid?
Posted by Paul van Delst on Thu, 12 Jul 2001 15:01:30 GMT
View Forum Message <> Reply to Message

Bill wrote:
>
> A better way to describe what could, but is not, implemented for IDL's keywords is a multise
> stage process
>
> 1. If a keyword on an invocation exactly matches a keyword in the functions's definition then it
> is that keyword, else
>
> 2. if the keyord on invocation is an abbreviation for exactly one keyword in the functions's
> definitition then it is that keyword, else
>
> 3. It is an error that can be determined statically.

I understand your point (and Jaco's and James'.... and Craig's too I think) but, to me,
the above rules are defined with only the programmer's (i.e. the person that wrote the
code that has some potential for ambiguousness (?) in the keywords) viewpoint.

My main, err, discomfort with allowing "The keyword is too short to be unique, therefore
it is unabbreviated" type of behaviour is that it does not take into account the person

who is using this code cold and is not an IDL whiz. I think that with a little bit of forethought, these issues can be eliminated by the code writer to save the puir wee unsuspecting future IDL user from some code that was written with potential ambiguous keyword problems. To paraphrase Reverend Lovejoy's wife: "will somebody *please* think of the users!" :o)

Phew.

O.k., no more poking pointy sticks at windmills for me. :o)

paulv

--
Paul van Delst A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
Fax:(301)763-8545 And drinking largely sobers us again.
Alexander Pope.

Subject: Re: Am I stupid?
Posted by david[2] on Thu, 12 Jul 2001 17:49:53 GMT
View Forum Message <> Reply to Message

Mark Hadfield writes:

>> I would be *extremely* happy if absolutely NO
>> keyword abbreviations were allowed.
>
> You must admit they are handy at the command line.

Alright, I'll concede they are handy at the command line. But as long as we are talking totally unrealistic expectations anyway, why not offer command completion at the IDL command line? That way those of us who haven't spent half our lives learning the secret runes of the EMACS editor would have something cool to play with, too.

But I think anyone who abbreviates keywords in code should be shocked as soon as his/her figures touch the compile button. And I think the compiler should barf all over him/her too. :-)

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: Am I stupid?
Posted by david[2] on Thu, 12 Jul 2001 17:56:35 GMT
View Forum Message <> Reply to Message

David Fanning writes:
>
> But I think anyone who abbreviates keywords in code
> should be shocked as soon as his/her figures touch
> the compile button. And I think the compiler should
> barf all over him/her too. :-)

You know, while we are talking about this, and
as long as I already have up a good head of steam...

The most grievous abusers of keyword abbreviations
are the good folks at RSI in the library code.
I can't tell you how many times that GROUP keyword
in XLOADCT has given me grief. :-(

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: Am I stupid?
Posted by R.Bauer on Thu, 19 Jul 2001 06:47:38 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:
>
> Bill wrote:
>>

>> A better way to describe what could, but is not, implemented for IDL's keywords is a multise
>> stage process
>>
>> 1. If a keyword on an invocation exactly matches a keyword in the functions's definition then it
>> is that keyword, else
>>
>> 2. if the keyord on invocation is an abbreviation for exactly one keyword in the functions's
>> definitition then it is that keyword, else
>>
>> 3. It is an error that can be determined statically.
>
> I understand your point (and Jaco's and James'.... and Craig's too I think) but, to me,
> the above rules are defined with only the programmer's (i.e. the person that wrote the
> code that has some potential for ambiguousness (?) in the keywords) viewpoint.
>
> My main, err, discomfort with allowing "The keyword is too short to be unique, therefore
> it is unabbreviated" type of behaviour is that it does not take into account the person
> who is using this code cold and is not an IDL whiz. I think that with a little bit of
> forethought, these issues can be eliminated by the code writer to save the puir wee
> unsuspecting future IDL user from some code that was written with potential ambiguous
> keyword problems. To paraphrase Reverend Lovejoy's wife: "will somebody *please* think of
> the users!" :o)
>
> Phew.
>
> O.k., no more poking pointy sticks at windmills for me. :o)

So this means no keywords only _extra and own rules to interpret
_extra as keywords.

Only a few new idl programmers will understand what's _extra is doing.

With one of the rsi internal routines we can create inside the routine
the variables
A and A1 and then they are not ambigous.

But what do we win and what not if we do so?

The routine itselfs has no keywords and normally these keyword names
well
defined and described and normally I see by this keywords what's the
routine is able to
do and what is missing.

A routine written in this way has different rules which are defined only
by the programmer.

Only a few special things will be better to use.

```
pro stupid,_extra=e
  tn=tag_names(e)
  print,tn[0]+'='+string(e.(0))
  print,tn[1]+'='+string(e.(1))
end
```

Please could we make a list which rsi idl library routines has these
problems.

regards
Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
http://www.fz-juelich.de/icg/icg1/
 ========== ======
a IDL library at ForschungsZentrum Juelich
 http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml

http://www.fz-juelich.de/zb/text/publikation/juel3786.html
 ========== ======

read something about linux / windows
http://www.suse.de/de/news/hotnews/MS.html

Subject: Re: Am I stupid?
Posted by Martin Schultz on Tue, 24 Jul 2001 09:25:15 GMT
View Forum Message <> Reply to Message

Paul van Delst <paul.vandelst@noaa.gov> writes:

> Bill wrote:
>>
>> A better way to describe what could, but is not, implemented for IDL's keywords is a multise
>> stage process
>>

>> 1. If a keyword on an invocation exactly matches a keyword in the functions's definition then it
>> is that keyword, else
>>
>> 2. if the keyord on invocation is an abbreviation for exactly one keyword in the functions's
>> definitition then it is that keyword, else
>>
>> 3. It is an error that can be determined statically.
>
> I understand your point (and Jaco's and James'.... and Craig's too I think) but, to me,
> the above rules are defined with only the programmer's (i.e. the person that wrote the
> code that has some potential for ambiguousness (?) in the keywords) viewpoint.
>
> My main, err, discomfort with allowing "The keyword is too short to be unique, therefore
> it is unabbreviated" type of behaviour is that it does not take into account the person
> who is using this code cold and is not an IDL whiz. I think that with a little bit of
> forethought, these issues can be eliminated by the code writer to save the puir wee
> unsuspecting future IDL user from some code that was written with potential ambiguous
> keyword problems. To paraphrase Reverend Lovejoy's wife: "will somebody *please* think of
> the users!" :o)
>
> Phew.
>
> O.k., no more poking pointy sticks at windmills for me. :o)
>
> paulv
>

Oh, how often have I asked that question in the subject to myself!

Maybe a little late to warm up this thread again, but with a topic that is
as near to my heart as keyword abbreviation, I just can't resist. What I would
like to propose is thinking a little out of the box and pick up the trends
that are out there in the software world. Why bother with ambiguities at all:
let the program decide what it wants to do with the keyword. Modern software
should just add three buttons to every program, one of which would have to be
activated after each computation: "Yes", "No", "Don't know". Hence, the user
"indicates" to the software if he or she is content with the result. If
so, the software will be happy and celebrate by taking a day off, if not,
the software will file this reply in its database and perhaps attempt to
learn from this experience in a future session (or version). If the user
continuously hits the "Don't know" button, he or she will automatically
be registered for a programming course.

Now, internally, this could probably be realized with an algorithm that
was recently developed by some smart mathematician in Hungary (or was it
Bulgaria ?): If a keyword has more than !PI letters, and is not one of the
words "TO", "I", "N", "GR" (this only for heritage reasons), "D", "T",
then the likelihood for a keyword that can be abbreviated is proportional

to the fifth power of the number of letters-!PI+1. Exceptions are: the specific keywords "TIME", "TIMEUNIT", and "TIMESTEP" which are treated seperately due to a user request. And because IDL shall also be usable by minors, certain words containing explicit language or violence are excluded from the vocabulary.

My final suggestion is a little more serious: Why not put everything related to TIME in a structure?

```
  Do_Something, TIME={value:0., unit:'years', step:1L }
```

has the virtue of unambiguity. If you use my little ChkStru program (available on http://www.mpimet.mpg.de/~schultz.martin/idl/) you can easily analyse this structure argument as in:

```
  IF ChkStru(time, 'value') then timevalue = time.value
  IF ChkStru(time, 'unit') then timeunit = time.unit
  IF ChkStru(time, 'step') then timestep = time.step
```

Altogether, I think there is clearly a need for RSI to properly define the line between IDL for programmers and IDL for scientists, and to try to make this distinction clear in future versions. A few key differences between these groups:

programmers:
*like variable declarations and hate type changes that are not asked for
*like fixed variable dimensions and hate secretly deleted trailing dimensions
*like as much syntax checking as possible at compile time

scientists:
*hate variable declarations and want the software to decide on the type
*hate limitations imposed by fixed variable dimensions
*hate error messages from the compiler and believe the program should
 know what they intend

Go figure!

Cheers,

Martin

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

Subject: Re: Am I stupid?
Posted by Paul van Delst on Tue, 24 Jul 2001 12:52:22 GMT
View Forum Message <> Reply to Message

Martin Schultz wrote:
>

> Altogether, I think there is clearly a need for RSI to properly define
> the line between IDL for programmers and IDL for scientists,.....

I thought they had by introducing object graphics? :o)

All that extra functionality..... sigh.

paulv

--
Paul van Delst A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
Fax:(301)763-8545 And drinking largely sobers us again.
Alexander Pope.

Subject: Re: Am I stupid?
Posted by colinr on Wed, 08 Aug 2001 11:12:01 GMT
View Forum Message <> Reply to Message

On 24 Jul 2001 11:25:15 +0200,
Martin Schultz <martin.schultz@dkrz.de> wrote:

> And because IDL shall also be usable
> by minors, certain words containing explicit language or violence are
> excluded from the vocabulary.

"MyPlot,X,Y,/FU" kind of thing?

--
Colin Rosenthal
Astrophysics Institute
University of Oslo
