## Subject: Re: Keyword abbreviation ambiguities Posted by Craig Markwardt on Sun, 04 Apr 2004 15:07:30 GMT View Forum Message <> Reply to Message

justspam03@yahoo.de (Oliver Thilmann) writes:

>> function testfunc, title=t, title1=t1

..

- > I don't see any ambiguity here. May I consider this a
- > (design-)bug, or is there some deeper meaning to this behaviour?

You may consider it a design bug -- I do -- but RSI does not. The ambiguity is that TITLE "could be" a shortened version of TITLE1. This has been discussed on the newsgroup, and most people agreed that the desired behavior is that if there is an exact keyword match, there should be no ambiguity error. Unfortunately, with the way that IDL presently works, there is no way for you to get your program working with those two keywords.

Craig	
,	EMAIL: craigmnet@REMOVEcow.physics.wisc.edu Derivatives   Remove "net" for better response

Subject: Re: Keyword abbreviation ambiguities
Posted by MKatz843 on Mon, 05 Apr 2004 01:10:12 GMT
View Forum Message <> Reply to Message

This bug most often bites me when I use really short keywords. For example, I like using "N" for the number of somethings, but doing so pretty much invalidates any other possible keyword beginning with the letter N. For example, in this procedure

pro test444, N=N, Napkin=Napkin print, 'April 4, 2004' end

calling IDL> test444, N=5

causes the aforementioned "% Ambiguous keyword abbreviation" error. I think we're stuck with this one.

M. Katz

View Forum Message <> Reply to Message

One way to overcome this would be to use the \_EXTRA keyword. One of many options would be

```
function testfunc, _EXTRA=e
;title=t,
;title1=t1
FOR i=0,N_TAGS(e)-1 DO $
CASE (TAG_NAMES(e))[i] OF
'TITLE': t = e.title
'TITLE1': t1 = e.title1
ENDCASE
print, t, t1
return, 0
end
```

Of course error checking should be performed. And full keywords should be used. For keyword completion, I think I once saw a post by Reimar Bauer on that topic.

```
"Oliver Thilmann" < justspam03@yahoo.de> schreef in bericht
news:c992bd37.0404040539.3446cff@posting.google.com...
> Hi.
>
> the following piece of code:
>
>> pro test
     y = testfunc( title=10, title1=20 )
>> end
>>
>> function testfunc, title=t, title1=t1
    print, t, t1
     return, 0
>>
>> end
>
> results in the error message (IDL6.0):
> % Ambiguous keyword abbreviation: TITLE.
> I don't see any ambiguity here. May I consider this a
> (design-)bug, or is there some deeper meaning to this behaviour?
> Regards
> Oliver
```

Subject: Re: Keyword abbreviation ambiguities

```
"mwvogel" <mvogel@rdiag.fgg.eur.nl> writes:
> One way to overcome this would be to use the _EXTRA keyword. One of many
> options would be
> function testfunc, _EXTRA=e
Greetings, it's true that one can use the _EXTRA keyword explicitly,
i.e. by directly passing a structure through EXTRA. However, the
EXTRA keyword processing still doesn't allow us to do the desired
thing, which is:
 testfunc, title='a', title1='b'
Yours.
Craig
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@REMOVEcow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
```

Subject: Re: Keyword abbreviation ambiguities Posted by mwvogel on Mon, 05 Apr 2004 11:47:40 GMT

View Forum Message <> Reply to Message

Dear All/Craig,

```
>> One way to overcome this would be to use the _EXTRA keyword. One of many
>> options would be
>> function testfunc, EXTRA=e
> ...
> Greetings, it's true that one can use the _EXTRA keyword explicitly,
> i.e. by directly passing a structure through _EXTRA. However, the
 EXTRA keyword processing still doesn't allow us to do the desired
> thing, which is:
   testfunc, title='a', title1='b'
Using the code below I get the following on my system
IDL> print, !VERSION
```

```
{ x86 Win32 Windows 5.4 Sep 25 2000 32 64} IDL> help, testfunc( title='a', title1='b') ab < Expression> INT = 0
```

I feel this code should work cross platform, but have not tested it. Of course Craig Markwardt and M. Katz were right pointing out that the explicit definition as originally requested will not work.

```
code:
function testfunc, _EXTRA=e
;title=t,
;title1=t1
FOR i=0,N_TAGS(e)-1 DO $
CASE (TAG_NAMES(e))[i] OF
'TITLE': t = e.title
'TITLE1': t1 = e.title1
ENDCASE
print, t, t1
return, 0
end
```

Subject: Re: Keyword abbreviation ambiguities Posted by R.Bauer on Mon, 05 Apr 2004 13:10:53 GMT

View Forum Message <> Reply to Message

```
mwvogel wrote:
> Dear All/Craig,
>
>>> One way to overcome this would be to use the _EXTRA keyword. One of many
>>> options would be
>>>
>>> function testfunc, EXTRA=e
>>
>> ...
>>
>> Greetings, it's true that one can use the _EXTRA keyword explicitly,
>> i.e. by directly passing a structure through _EXTRA. However, the
>> _EXTRA keyword processing still doesn't allow us to do the desired
>> thing, which is:
>>
>> testfunc, title='a', title1='b'
>
> Using the code below I get the following on my system
> IDL> print, !VERSION
```

```
> { x86 Win32 Windows 5.4 Sep 25 2000
                                              32
                                                    64}
> IDL> help, testfunc( title='a', title1='b')
> ab
> <Expression> INT
                                  0
> I feel this code should work cross platform, but have not tested it. Of
> course Craig Markwardt and M. Katz were right pointing out that the explicit
  definition as originally requested will not work.
> code:
  function testfunc, _EXTRA=e
    :title=t.
    :title1=t1
>
    FOR i=0,N_TAGS(e)-1 DO $
>
     CASE (TAG_NAMES(e))[i] OF
     'TITLE': t = e.title
>
     'TITLE1': t1 = e.title1
>
     ENDCASE
>
    print, t, t1
>
    return, 0
>
   end
>
This does only work if the keywords are completly written.
It would be bad to be forced to do it.
If you like to have different titles it is better to use different names
,e.g. legend_title, title, page_title.
You could group by a prefix like legend_ a whole bunch of keywords which
 then could be extracted from the _extra structure easily and send to a
specific routine as e.g. title and so on.
I am using in this case extract_extra() from our library.
result=extract_extra(extra,'legend_*')
regards
Reimar
Reimar Bauer
```

Institut fuer Stratosphaerische Chemie (ICG-I)

Forschungszentrum Juelich email: R.Bauer@fz-juelich.de

-----

a IDL library at ForschungsZentrum Juelich http://www.fz-juelich.de/icg/icg-i/idl\_icglib/idl\_lib\_intro. html

\_\_\_\_\_

Subject: Re: Keyword abbreviation ambiguities
Posted by R.Bauer on Mon, 05 Apr 2004 13:38:10 GMT
View Forum Message <> Reply to Message

## mwvogel wrote:

- > One way to overcome this would be to use the \_EXTRA keyword. One of many
- > options would be

>

- > function testfunc, \_EXTRA=e
- > ;title=t,
- > :title1=t1
- > FOR i=0,N\_TAGS(e)-1 DO \$
- > CASE (TAG\_NAMES(e))[i] OF
- > 'TITLE': t = e.title
- > 'TITLE1': t1 = e.title1
- > ENDCASE
- > print, t, t1
- > return, 0
- > end

>

- > Of course error checking should be performed. And full keywords should be
- > used. For keyword completion, I think I once saw a post by Reimar Bauer on
- > that topic.

This was a long thread

http://groups.google.de/groups?q=\_extra+Reimar+group:comp.la ng.idl-pvwave&hl=de&lr=&ie=UTF-8&group=comp. lang.idl-pvwave&selm=3B56828A.71B9D015%40fz-juelich.de&a mp;rnum=3

It is a way to get out of this rules but it is not quite easy to explain someone how to call a routine which is written this way.

routine\_info gives no result about keywords. auto writing of header pages are impossible.

Until now we have a lot of routines where keyword inhiterance is used. And it is quite easy to find the routine where the keywords are explained and the others could use \_extra, \_strict\_extra or \_ref\_extra.

## Reimar

```
"Oliver Thilmann" <justspam03@yahoo.de> schreef in bericht
> news:c992bd37.0404040539.3446cff@posting.google.com...
>> Hi,
>>
>> the following piece of code:
>>
>>> pro test
>>> y = testfunc( title=10, title1=20 )
>>> end
>>>
>>> function testfunc, title=t, title1=t1
>>> print, t, t1
>>> return, 0
>>> end
>>
>> results in the error message (IDL6.0):
>> % Ambiguous keyword abbreviation: TITLE.
>>
>> I don't see any ambiguity here. May I consider this a
>> (design-)bug, or is there some deeper meaning to this behaviour?
>> Regards
>> Oliver
>
>
Reimar Bauer
Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
     a IDL library at ForschungsZentrum Juelich
 http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html
```