Subject: Re: Pass-By-Reference question.
Posted by Chris Lee on Wed, 10 Mar 2004 09:37:35 GMT
View Forum Message <> Reply to Message

In article <405594fa.0403100119.34769176@posting.google.com>, "Tim Robishaw" <timrobishaw@yahoo.com> wrote:

> Hi there,
> I've been using _REF_EXTRA for a long time now and have come to
> appreciate that values of keywords sent to a module with the _REF_EXTRA
> mechanism will override any keywords that are set inside the module.
> Here is a great example: pro whaddup, _REF_EXTRA=_extra
> plot, [0], [0], /NODATA, XRANGE=[-5,5], YRANGE=[-5,5], PSYM=4,
> _EXTRA=_extra
> end
> IDL> whaddup
> I see what the module asked for: axes from -5 to 5 with no datum
> plotted.
> Now I can override the XRANGE and YRANGE keyword values by passing them
> by reference...
> IDL> whaddup, XRANGE=[-1,1], YRANGE=[-1,1] I see what I asked for: axes
> from -1 to 1. However, if I send NODATA set to ZERO, i.e., I'd really
> like to see my datum this time...
> IDL> whaddup, XRANGE=[-1,1], YRANGE=[-1,1], NODATA=0 I don't get what I
> asked for. The value for NODATA sent by reference does not override the
> value set inside the module. This is also true for the /NOERASE
> keyword.
> IDL newsgroup: whaddup?

Hi,

IDL> plot, findgen(10), nodata=1
IDL> plot, findgen(10), nodata=1,nodata=0
% Duplicate keyword NODATA in call to: PLOT
% Execution halted at: $MAIN$
IDL> e={nodata:0}
IDL> plot, findgen(10), nodata=1,_EXTRA=e

Duplicate keywords give IDL no chance of knowing what to do, hence it complains, and it only looks in the _EXTRA struct if it doesn't find what it wants elsewhere. If you want an option of overriding the nodata keyword, you need to look for it first, hence

```
pro whaddup, nodata=nodata,_REF_EXTRA=_extra

nodata_internal = 0
```

```
if(arg_present(nodata)) then  nodata_internal=nodata > 0 < 1

plot, [0], [0], NODATA=nodata_internal, XRANGE=[-5,5], YRANGE=[-5,5], PSYM=4,
_EXTRA=_extra

end
```

This works for any boolean keyword, for multi valued /array keywords you need N_ELEMENTS and IF..THEN..ELSE, depending on the default conditions you want for the variable.

Chris.

Subject: Re: Pass-By-Reference question.
Posted by Mark Hadfield on Wed, 10 Mar 2004 20:13:25 GMT
View Forum Message <> Reply to Message

Christopher Lee wrote:
> In article <405594fa.0403100119.34769176@posting.google.com>, "Tim
> Robishaw" <timrobishaw@yahoo.com> wrote:
>
>> Hi there,
>> I've been using _REF_EXTRA for a long time now and have come to
>> appreciate that values of keywords sent to a module with the _REF_EXTRA
>> mechanism will override any keywords that are set inside the module.
>> Here is a great example: pro whaddup, _REF_EXTRA=_extra
>> plot, [0], [0], /NODATA, XRANGE=[-5,5], YRANGE=[-5,5], PSYM=4,
>> _EXTRA=_extra
>> end
>> IDL> whaddup
>> I see what the module asked for: axes from -5 to 5 with no datum
>> plotted.
>> Now I can override the XRANGE and YRANGE keyword values by passing them
>> by reference...
>> IDL> whaddup, XRANGE=[-1,1], YRANGE=[-1,1] I see what I asked for: axes
>> from -1 to 1. However, if I send NODATA set to ZERO, i.e., I'd really
>> like to see my datum this time...
>> IDL> whaddup, XRANGE=[-1,1], YRANGE=[-1,1], NODATA=0 I don't get what I
>> asked for. The value for NODATA sent by reference does not override the
>> value set inside the module. This is also true for the /NOERASE
>> keyword.
>
> Hi,
>
> IDL> plot, findgen(10), nodata=1
> IDL> plot, findgen(10), nodata=1,nodata=0
> % Duplicate keyword NODATA in call to: PLOT

> % Execution halted at: $MAIN$
> IDL> e={nodata:0}
> IDL> plot, findgen(10), nodata=1,_EXTRA=e
>
> Duplicate keywords give IDL no chance of knowing what to do, hence it
> complains, and it only looks in the _EXTRA struct if it doesn't find what
> it wants elsewhere.

Not so. As Tim said, keywords passed in via inheritance override
keywords supplied explicitly. This is a very useful feature and
virtually all of my code relies on it.

In previous versions of IDL there have been glitches in IDL's
implementation of keyword precedence (for a while it worked with
value-inheritance but not reference-inheritance) but it has worked the
way I describe since (I think) 5.4 or 5.5. There have been extensive,
confusing discussions about this on the newsgroup in the past.

> If you want an option of overriding the nodata
> keyword, you need to look for it first, hence
>
> pro whaddup, nodata=nodata,_REF_EXTRA=_extra
>
> nodata_internal = 0
> if(arg_present(nodata)) then nodata_internal=nodata > 0 < 1
>
> plot, [0], [0], NODATA=nodata_internal, XRANGE=[-5,5], YRANGE=[-5,5], PSYM=4,
> _EXTRA=_extra
>
> end

Having inherited keywords override explicit keywords avoids the need for
all this and allows much cleaner code.

As to why Tim's example works for XRANGE & YRANGE but not NODATA, I have
no idea. Perhaps you are not really doing what you think you are doing.

--
Mark Hadfield "Ka puwaha te tai nei, Hoea tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

*****

Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.
Gratuitous text added to subvert silly news server restrictions.

Subject: Re: Pass-By-Reference question.
Posted by Craig Markwardt on Wed, 10 Mar 2004 21:24:27 GMT
View Forum Message <> Reply to Message

Mark Hadfield <m.hadfield@niwa.co.nz> writes:
>
> Not so. As Tim said, keywords passed in via inheritance override
> keywords supplied explicitly. This is a very useful feature and
> virtually all of my code relies on it.
>

Mark, are you relying on a documented feature of IDL, or are you just assuming that because it works, it's right? I've never assumed that keyword inheritance works by overriding explicit values. In fact, I would have assumed the exact opposite, namely that the explicit overrides the generic.

> As to why Tim's example works for XRANGE & YRANGE but not NODATA, I have
> no idea. Perhaps you are not really doing what you think you are doing.

I agree, the Tim's example is truly strange.

Craig

Subject: Re: Pass-By-Reference question.
Posted by timrobishaw on Wed, 10 Mar 2004 23:34:46 GMT
View Forum Message <> Reply to Message

"Christopher Lee" <cl@127.0.0.1> wrote in message
> Duplicate keywords give IDL no chance of knowing what to do, hence it
> complains, and it only looks in the _EXTRA struct if it doesn't find what
> it wants elsewhere. If you want an option of overriding the nodata
> keyword, you need to look for it first

I agree that the best thing to do is to pass NODATA in as an explicit keyword in this case. However, I strongly disagree with your interpretation that IDL is confused by duplicate keywords in this scenario: the point is that keywords passed by reference are advertised to override keywords set inside the module (and this is a great thing since you can have a default set of keywords that can be superseded by user input!) In fact, in the example initially provided, the XRANGE and YRANGE keywords are "duplicated" by passing them in by reference and, as advertised, the values passed by reference override the values set in the call to plot. Anyone have any other ideas?

Subject: Re: Pass-By-Reference question.
Posted by Mark Hadfield on Wed, 10 Mar 2004 23:55:17 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:
> Mark Hadfield <m.hadfield@niwa.co.nz> writes:
>
>> Not so. As Tim said, keywords passed in via inheritance override
>> keywords supplied explicitly. This is a very useful feature and

>> virtually all of my code relies on it.
>>
>
>
> Mark, are you relying on a documented feature of IDL, or are you just
> assuming that because it works, it's right? I've never assumed that
> keyword inheritance works by overriding explicit values. In fact, I
> would have assumed the exact opposite, namely that the explicit
> overrides the generic.

This behaviour *is* documented for value inheritance, eg under the topic "Keyword Inheritance", heading "Writing a wrapper routine", sub-heading "By Value" the IDL 6.0 help file says the following"

> Note that keywords passed into a routine via _EXTRA override previous settings of that keyword. For example, the call:
>
> PLOT, a, b, COLOR = color, _EXTRA = {COLOR: 12}
>
> specifies a color index of 12 to PLOT.

Under the sub-heading "By Reference" it says something very similar, but I must admit I don't understand why structures get introduced here:

> These inherited keywords are then passed from TEST to the PLOT routine using the _EXTRA keyword. Note that keywords passed into a routine via _EXTRA override previous settings of that keyword. For example, the call:
>
> PLOT, a, b, COLOR = color, _EXTRA = {COLOR: 12}
>
> specifies a color index of 12 to PLOT. Also note that we are passing a structure (the by value format used by _EXTRA) as the value of the extra keyword to a routine that uses the by reference keyword inheritance mechanism (_REF_EXTRA). There is no problem in doing this, because each routine establishes its own inheritance mechanism independent of any other routines that may be calling it. However, any keyword values that are changed within PLOT will fail to be returned to the caller due to the use of the by-value mechanism.

Anyway, back in Aug 2000, during the beta test period of IDL 5.4, there was a long thread on this newsgroup with the title "Keyword precedence", involving me & JD mainly, during which I noted that there was an inconsistency between the keyword precedence rules for value and reference inheritance, and I argued that they should both behave the way described above (inherited keywords override explicit keywords). I also raised this with RSI and persuaded them to change the behaviour for reference inheritance by the time IDL 5.4 final was released.

If you review that thread (though I doubt you'll have the stamina to read the whole thing) you may note that I began by saying that I had until recently assumed the contrary (explicit should override inherited). My first message in that thread explains why I changed my mind: it allows one tow write cleaner wrapper routines.

--
Mark Hadfield "Ka puwaha te tai nei, Hoea tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: Pass-By-Reference question.
Posted by timrobishaw on Thu, 11 Mar 2004 03:04:43 GMT
View Forum Message <> Reply to Message

Craig Markwardt <craigmnet@REMOVEcow.physics.wisc.edu> wrote
> Mark, are you relying on a documented feature of IDL, or are you just
> assuming that because it works, it's right? I've never assumed that
> keyword inheritance works by overriding explicit values. In fact, I
> would have assumed the exact opposite, namely that the explicit
> overrides the generic.

Craig, the online IDL help document explicitly details this behavior as a feature of passing by reference.

IDL> ? _REF_EXTRA

>> As to why Tim's example works for XRANGE & YRANGE but not NODATA, I have
>> no idea. Perhaps you are not really doing what you think you are doing.
>
> I agree, the Tim's example is truly strange.

Oh, no, the Tim knows what he's doing. ;-> I'm not going to go into what brought this up, but as for being strange, I think the example I provided is perfectly legit: You might want the default state to have a plot's axes run from -5 to 5 (forget about the specifics, just imagine that would be the routine's default state.) However, a user might want to zoom in or out and therefore would like to control the XRANGE via the keyword inheritance mechanism. Makes sense to me. Works like a charm. I'd just like to know why it doesn't work for NODATA and NOERASE.

Subject: Re: Pass-By-Reference question.

Tim Robishaw wrote:
> Craig Markwardt <craigmnet@REMOVEcow.physics.wisc.edu> wrote
>
>> Mark, are you relying on a documented feature of IDL, or are you just
>> assuming that because it works, it's right? I've never assumed that
>> keyword inheritance works by overriding explicit values. In fact, I
>> would have assumed the exact opposite, namely that the explicit
>> overrides the generic.
>
>
> Craig, the online IDL help document explicitly details this behavior as
> a feature of passing by reference.
>
> IDL> ? _REF_EXTRA
>
>
>>> As to why Tim's example works for XRANGE & YRANGE but not NODATA, I have
>>> no idea. Perhaps you are not really doing what you think you are doing.
>>
>> I agree, the Tim's example is truly strange.
>
>
> Oh, no, the Tim knows what he's doing. ;-> I'm not going to go into
> what brought this up, but as for being strange, I think the example I
> provided is perfectly legit:

I don't think Craig was suggesting that you are trying to do something strange(*) just that IDL's behaviour, as reported by you, is strange. Perhaps it's a bug in PLOT?

To rule out the possibility that you have made some basic mistake, can you please post some self-contained test code that illustrates the problem.

(*)Though if you were doing something strange with IDL, you certainly wouldn't be the first. Ask David. Ask David's wife.

--
Mark Hadfield "Ka puwaha te tai nei, Hoea tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

***
More included text than new. Yada yada yada. More included text than new. Yada yada yada. More included text than new. Yada yada yada. More included text than new. Yada yada yada. More included text than new. Yada yada yada. More included text than new. Yada yada yada. More

included text than new. Yada yada yada. More included text than new. Yada yada yada.

Subject: Re: Pass-By-Reference question.
Posted by David Fanning on Thu, 11 Mar 2004 04:34:41 GMT
View Forum Message <> Reply to Message

Mark Hadfield writes:

> Ask David. Ask David's wife.

Don't ask my wife. She doesn't want to hear about IDL anymore. :-(

Cheers,

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

Subject: Re: Pass-By-Reference question.
Posted by Timm Weitkamp on Thu, 11 Mar 2004 09:28:35 GMT
View Forum Message <> Reply to Message

Today at 17:24 +1300, Mark Hadfield wrote:

> [...]
>

> I don't think Craig was suggesting that you are trying to do something
> strange(*) just that IDL's behaviour, as reported by you, is strange.
> Perhaps it's a bug in PLOT?
>
> To rule out the possibility that you have made some basic mistake, can
> you please post some self-contained test code that illustrates the problem.

I think the little example that came with Tim's first posting is
self-contained enough. I ran it with 5.4 and 6.0 and with both versions
got the same result that he reports.

Here's an observation that makes it even stranger: when you change
"whaddup.pro" so that NODATA is explicitly set to 0 (instead of one)
inside the routine, and you then try to override that by calling

```
IDL> whaddup, nodata=1
```

The amazing thing is that this way it works (i.e., inherited overrides
explicit keyword).

Timm

--
Timm Weitkamp <http://people.web.psi.ch/weitkamp>

Subject: Re: Pass-By-Reference question.
Posted by Chris Lee on Thu, 11 Mar 2004 10:12:24 GMT
View Forum Message <> Reply to Message

In article <405594fa.0403101534.55e8403c@posting.google.com>, "Tim
Robishaw" <timrobishaw@yahoo.com> wrote:

> "Christopher Lee" <cl@127.0.0.1> wrote in message
>> Duplicate keywords give IDL no chance of knowing what to do, hence it
>> complains, and it only looks in the _EXTRA struct if it doesn't find
>> what it wants elsewhere. If you want an option of overriding the nodata
>> keyword, you need to look for it first
> I agree that the best thing to do is to pass NODATA in as an explicit
> keyword in this case. However, I strongly disagree with your
> interpretation that IDL is confused by duplicate keywords in this
> scenario: the point is that keywords passed by reference are advertised
> to override keywords set inside the module (and this is a great thing
> since you can have a default set of keywords that can be superseded by
> user input!) In fact, in the example initially provided, the XRANGE and
> YRANGE keywords are "duplicated" by passing them in by reference and, as
> advertised, the values passed by reference override the values set in

> the call to plot. Anyone have any other ideas?

plot, findgen(10),nodata=0, _EXTRA={nodata:1}
plot, findgen(10), nodata=1, _EXTRA={nodata:0}

both product the same result.

contour, dist(100),/follow, /downhill, _EXTRA={follow:0,downhill:0}
contour, dist(100),follow=0, downhill=0, _EXTRA{=follow:1,downhill:1}

and again...

help, make_array(dimension=[2],value=0.0,double=1,_EXTRA={double:0 })
<Expression> DOUBLE = Array[2]
IDL> help, make_array(dimension=[2],value=0.0,double=0,_EXTRA={double:1 })
<Expression> DOUBLE = Array[2]

and again...

IDL isn't confused?

It has something to do with built in functions and boolean input
variables (or at least, inputs that are considered boolean,
where the keyword is usually set as /flag).

It doesn't appear to work with user functions, in any combination that I
tried.

Chris.

Subject: Re: Pass-By-Reference question.
Posted by David Fanning on Thu, 11 Mar 2004 12:40:49 GMT
View Forum Message <> Reply to Message

Timm Weitkamp writes:

> Here's an observation that makes it even stranger: when you change
> "whaddup.pro" so that NODATA is explicitly set to 0 (instead of one)
> inside the routine, and you then try to override that by calling
>
> IDL> whaddup, nodata=1
>
> The amazing thing is that this way it works (i.e., inherited overrides
> explicit keyword).

I discovered this yesterday when I was playing around with
the problem. It reminds me of what happens sometimes when
you incorrectly use KEYWORD_SET instead of N_ELEMENTS to
determine if a keyword is defined or not, and set a
default value. For example, if you use this expression:

```
  IF Keyword_Set(mykeyword) THEN mykeyword = 3
```

instead of this expression:

```
   IF N_ELements(mykeyword) EQ 0 THEN mykeyword = 3
```

then it is impossible to set this keyword to 0. I just
haven't been able to come up with a specific mechanism
for this particular situation. But I *have* seen RSI
programmers make this kind of mistake in code, so I think
the theory is plausible.

Cheers,

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

Subject: Re: Pass-By-Reference question.
Posted by Mark Hadfield on Thu, 11 Mar 2004 20:07:45 GMT
View Forum Message <> Reply to Message

Christopher Lee wrote:
>
> plot, findgen(10),nodata=0, _EXTRA={nodata:1}
> plot, findgen(10), nodata=1, _EXTRA={nodata:0}
>
> both produce the same result.
>
> contour, dist(100),/follow, /downhill, _EXTRA={follow:0,downhill:0}
> contour, dist(100),follow=0, downhill=0, _EXTRA{=follow:1,downhill:1}
>
> and again...
>
>   help, make_array(dimension=[2],value=0.0,double=1,_EXTRA={double:0 })
> <Expression>   DOUBLE   = Array[2]
> IDL> help, make_array(dimension=[2],value=0.0,double=0,_EXTRA={double:1 })
> <Expression>   DOUBLE   = Array[2]

Nice work. These occur in 6.0 and 5.6, which are the only versions I have.

It's a bug. Who's going to report it?

--
Mark Hadfield "Ka puwaha te tai nei, Hoea tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: Pass-By-Reference question.
Posted by Craig Markwardt on Fri, 12 Mar 2004 01:37:22 GMT
View Forum Message <> Reply to Message

Mark Hadfield <m.hadfield@niwa.co.nz> writes:

> Christopher Lee wrote:
...
>> help, make_array(dimension=[2],value=0.0,double=1,_EXTRA={double:0 })
>> <Expression> DOUBLE = Array[2]
>> IDL> help, make_array(dimension=[2],value=0.0,double=0,_EXTRA={double:1 })
>> <Expression> DOUBLE = Array[2]
>
> Nice work. These occur in 6.0 and 5.6, which are the only versions I have.
>
> It's a bug. Who's going to report it?

It's not like I can pretend to know what's going on, as my previous
post already evidenced. But, I can say that the above MAKE_ARRAY()
behavior exists in IDL 5.5, 5.4, 5.2, 5.1, 5.0, and 4.0.1!!!

Craig

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

Subject: Pass-By-Reference question. (ARG_PRESENT)
Posted by Andry William (Please on Fri, 12 Mar 2004 19:52:51 GMT
View Forum Message <> Reply to Message

> pro whaddup, nodata=nodata,_REF_EXTRA=_extra
>
> nodata_internal = 0

>   if(arg_present(nodata)) then nodata_internal=nodata > 0 < 1
>

Hi All,

This might be off topic but can somebody explain to me why the
following statements give different result. (Maybe I did not really
understand the IDL manual). (I am using IDL 6.0).

I have the following procedure:

```
PRO t_arg_present,arg1=arg1

IF KEYWORD_SET(arg1) THEN PRINT, "'arg1' is set to ", arg1
IF ARG_PRESENT(arg1) THEN PRINT, "'arg1' is present", arg1

END
```

If I do:
IDL> t_arg_present,arg1=1
It returns:
'arg1' is set to   1

If I do:
IDL> t_arg_present,arg1=0
Nothing comes out.

If I do:
IDL> a=0
IDL> t_arg_present,arg1=a
It returns:
'arg1' is present   0

If I do:
IDL> a=1
IDL> t_arg_present,arg1=a
It returns:
'arg1' is set to   1
'arg1' is present   1

Are all the output the expected one?

Thanks,

Andry