
Subject: _extra keyword

Posted by [antonioolita](#) on Fri, 01 Apr 2005 10:12:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

How Must I use this keyword?

If I have a procedure

```
PRO TRY, a,b,_extra=extra
```

```
    tv, c
```

```
END
```

where the programmer tell me that for extra is available every keyword
of (for example) PLOT,

Can I call simply

```
TRY, a, b, thick=3
```

where thick is a valid keyword for PLOT(and therefore for TRY)...

or not?

Thanks for the answers

Antonio

Subject: Re: _extra keyword

Posted by [David Fanning](#) on Thu, 07 Apr 2005 12:33:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

Antonio writes:

- > Yes I think that's I want but....
- > but in the procedure that I try to use this keyword seems not to run.
- > The compiler don't made error message but there's no evident effect of
- > this keyword in my plot (a map).
- > More simply:
- > if I have a _extra keyword in a procedure how can I use it?
- > I can use EVERY other keyword of the procedure associated to them?

The problem (and beauty) of _EXTRA is that unknown keywords are silently ignored by the procedures that use this mechanism. In practice, this means that if you misspell a keyword, no one ever complains, but the keyword has no effect, either. I strongly suspect that is your problem.

One way to handle this problem is to use the _STRICT_EXTRA

keyword (in place of `_EXTRA`) on the last procedure or function in the keyword inheritance chain. The `_STRICT_EXTRA` *will* complain if it gets a keyword it doesn't understand, so you can use this to trap for misspelled keywords.

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: `_extra` keyword

Posted by [Michael Wallace](#) on Thu, 07 Apr 2005 19:52:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

- > The problem (and beauty) of `_EXTRA` is that unknown
- > keywords are silently ignored by the procedures that
- > use this mechanism. In practice, this means that if you
- > misspell a keyword, no one ever complains, but the keyword
- > has no effect, either. I strongly suspect that is your
- > problem.
- >
- > One way to handle this problem is to use the `_STRICT_EXTRA`
- > keyword (in place of `_EXTRA`) on the last procedure or function
- > in the keyword inheritance chain. The `_STRICT_EXTRA` *will*
- > complain if it gets a keyword it doesn't understand, so you
- > can use this to trap for misspelled keywords.

Just as a programming style note, when I'm in a routine which only passes extra keywords to one other routine, I always use `_STRICT_EXTRA`.

That guarantees me that if I mistype something or get a wrong keyword, the error will be caught.

The only time I don't use `_STRICT_EXTRA` is if I'm going to pass the extra arguments to at least two other routines. In this case you'd want the first routine to ignore the keywords associated with the second and vice versa. However, I only recommend doing this when it's really necessary. I've found that extra keywords are much easier to debug and maintain when you're able to use `_STRICT_EXTRA`.

-Mike

Subject: Re: _extra keyword
Posted by [David Fanning](#) on Thu, 07 Apr 2005 19:58:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

Michael Wallace writes:

- > Just as a programming style note, when I'm in a routine which only
- > passes extra keywords to one other routine, I always use _STRICT_EXTRA.
- > That guarantees me that if I mistype something or get a wrong
- > keyword, the error will be caught.
- >
- > The only time I don't use _STRICT_EXTRA is if I'm going to pass the
- > extra arguments to at least two other routines. In this case you'd want
- > the first routine to ignore the keywords associated with the second and
- > vice versa. However, I only recommend doing this when it's really
- > necessary. I've found that extra keywords are much easier to debug and
- > maintain when you're able to use _STRICT_EXTRA.

Amen to this!

Cheers,

David

--

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

Subject: Re: _EXTRA KEYWORD
Posted by [David Fanning](#) on Wed, 04 Nov 2009 23:22:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dandan writes:

- > Dose _EXTRA pass the keywords from the caller to called routine or the
- > opposite?

_EXTRA allows you to call a routine with keywords that were not defined for the routine. Usually, that routine uses those keywords in a call to some other routine, but what you do with them once they are packaged up in the extra structure is up to you.

Cheers,

David

Subject: Re: _EXTRA KEYWORD
Posted by [dandan](#) on Thu, 05 Nov 2009 07:37:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Nov 4, 3:22 pm, David Fanning <da...@dfanning.com> wrote:

> Dandan writes:
>> Dose _EXTRA pass the keywords from the caller to called routine or the
>> opposite?
>
> _EXTRA allows you to call a routine with keywords that were not
> defined for the routine. Usually, that routine uses those keywords
> in a call to some other routine, but what you do with them once they
> are packaged up in the extra structure is up to you.
>
> Cheers,
>
> David

Thank you very much!

Subject: Re: _EXTRA KEYWORD
Posted by [mankoff](#) on Wed, 11 Nov 2009 22:47:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Nov 9, 1:09 am, Reimar Bauer <R.Ba...@fz-juelich.de> wrote:

> dandan schrieb:
>
>> Hi, everyone,
>> Dose _EXTRA pass the keywords from the caller to called routine or the
>> opposite?
>
> You may be also interested into the _REF_EXTRA keyword.
>
> The presence of a name in the _REF_EXTRA value indicates that a keyword
> of that name was passed, and its value is available to be passed on in a
> function or procedure call (using either _EXTRA or _STRICT_EXTRA).
>
> cheers
> Reimar

I still have not entirely wrapped my head around _EXTRA, _REF_EXTRA,
and _STRICT_EXTRA despite numerous readings of discussions on this
group about these keywords.

Does anyone have a graphical explanation of these? A flow-chart with
arrows and diamonds and YES/NO keywords perhaps? I have a feeling that
would get the concept across. If not, perhaps I'll take what I do

understand of these keywords and try to create such a chart.

-k.

Subject: Re: `_EXTRA` KEYWORD

Posted by [ben.bighair](#) on Thu, 12 Nov 2009 00:23:59 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Nov 11, 5:47 pm, mankoff <mank...@gmail.com> wrote:

> On Nov 9, 1:09 am, Reimar Bauer <R.Ba...@fz-juelich.de> wrote:

>

>> dandan schrieb:

>

>>> Hi, everyone,

>>> Dose `_EXTRA` pass the keywords from the caller to called routine or the

>>> opposite?

>

>> You may be also interested into the `_REF_EXTRA` keyword.

>

>> The presence of a name in the `_REF_EXTRA` value indicates that a keyword

>> of that name was passed, and its value is available to be passed on in a

>> function or procedure call (using either `_EXTRA` or `_STRICT_EXTRA`).

>

>> cheers

>> Reimar

>

> I still have not entirely wrapped my head around `_EXTRA`, `_REF_EXTRA`,

> and `_STRICT_EXTRA` despite numerous readings of discussions on this

> group about these keywords.

>

> Does anyone have a graphical explanation of these? A flow-chart with

> arrows and diamonds and YES/NO keywords perhaps? I have a feeling that

> would get the concept across. If not, perhaps I'll take what I do

> understand of these keywords and try to create such a chart.

>

> -k.

Hi,

I am not sure if this analogy is correct, but it works for my limited use. Assume that `_EXTRA` and `_REF_EXTRA` are fishing lines.

`_REF_EXTRA` has bait and a hook so you can pull things back. A

`_REF_EXTRA` is passed by reference - so you can send in a value but if

it is modified (like an output keyword) then the reference will

reflect that change. This is how you can land the trout, so to

speak.

On the other hand, `_EXTRA` has bait but no hook so you can send things down in but not get them back. That is because `_EXTRA` is passed by value - any modifications the trout makes under water are hidden from you.

`_STRICT_EXTRA` is just like `_EXTRA`, but if you use the wrong kind of bait the trout will complain (or issue an error and stop if they know any IDL).

Even if my analogy is correct, it might be too simplistic to be complete. Hope it helps.

Cheers,
Ben

Subject: Re: `_EXTRA` KEYWORD
Posted by [David Fanning](#) on Thu, 12 Nov 2009 01:36:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

ben.bighair writes:

> Even if my analogy is correct, it might be too simplistic to be
> complete. Hope it helps.

I would just add that `_EXTRA` and `_REF_EXTRA` are always used on the boat (the procedure or function definition line), and `_STRICT_EXTRA` is always used at the end of the line. ;-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: `_EXTRA` KEYWORD
Posted by [Michael Galloy](#) on Thu, 12 Nov 2009 02:54:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> ben.bighair writes:

>

>> Even if my analogy is correct, it might be too simplistic to be

>> complete. Hope it helps.

>

> I would just add that `_EXTRA` and `_REF_EXTRA` are always

> used on the boat (the procedure or function definition

> line), and `_STRICT_EXTRA` is always used at the end

> of the line. ;-)

Well, I would say you have the choice of `_EXTRA` and `_REF_EXTRA` on the boat and the choice between `_EXTRA` and `_STRICT_EXTRA` on the line.

Mike

--

www.michaelgalloy.com

Research Mathematician

Tech-X Corporation

Subject: Re: `_EXTRA` KEYWORD

Posted by [David Fanning](#) on Thu, 12 Nov 2009 03:01:18 GMT

[View Forum Message](#) <> [Reply to Message](#)

Michael Galloy writes:

> Well, I would say you have the choice of `_EXTRA` and `_REF_EXTRA` on the

> boat and the choice between `_EXTRA` and `_STRICT_EXTRA` on the line.

You can have as many on the line as you like (and your license allows), but the `_STRICT_EXTRA` better be on the **end** of the line if it is going to do any good.

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")