
Subject: Re: Unsupported keyword on older IDL version
Posted by [thompson](#) on Mon, 29 Jul 2002 14:40:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paul Hick <pphick@ucsd.edu> writes:

> This kind of problem must have come up in this group before, but here it
> is anyways.

> If I call the IDL systime function with the utc keyword:
> print, systime(/utc)

> on IDL versions older than 5.4 I get the expected 'keyword not
> supported' error message, since, well, it's not supported, right? The
> error message pops up when the function containing the systime call is
> compiled, so testing e.g. !version.release before executing the systime
> call doesn't work.

> The only solution I have come up with so far is to put the systime call
> inside a call_function call:
> print, call_function('systime', /utc)
> This will compile (I will still need to test for !version.release before
> actually executing the statement of course).

> This to me looks like a somewhat illegitimate use of call_function. Not
> that I mind bending the rules a bit, but are there any other (better)
> techniques of dealing with keywords that don't work in all IDL versions.

Why do you call this "illegitimate"? To me this seems like just the sort of
thing that call_function() was designed for. I've always done it this way in
the past. In fact, I've usually used execute() in the past, but
call_function() is more elegant.

Bill Thompson

Subject: Re: Unsupported keyword on older IDL version
Posted by [Craig Markwardt](#) on Mon, 29 Jul 2002 15:14:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paul Hick <pphick@ucsd.edu> writes:

> This kind of problem must have come up in this group before, but here it
> is anyways.

>
> If I call the IDL systime function with the utc keyword:
> print, systime(/utc)

>
> on IDL versions older than 5.4 I get the expected 'keyword not
> supported' error message, since, well, it's not supported, right? The
> error message pops up when the function containing the systime call is
> compiled, so testing e.g. !version.release before executing the systime
> call doesn't work.
>
> The only solution I have come up with so far is to put the systime call
> inside a call_function call:
> print, call_function('systime', /utc)
> This will compile (I will still need to test for !version.release before
> actually executing the statement of course).

Another possibility is to use the _EXTRA mechanism:

```
;; Assume EXTRA is undefined  
if double(!version.release) GE 5.4 then $  
  extra = {utc: 1}  
value = systime(_EXTRA=extra)
```

Anyway you cut it, you will need an IDL version check.

Craig

--

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

Subject: Re: Unsupported keyword on older IDL version
Posted by [David Fanning](#) on Mon, 29 Jul 2002 15:34:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Craig Markwardt (craigmnet@cow.physics.wisc.edu) writes:

> Another possibility is to use the _EXTRA mechanism:
>
> ;; Assume EXTRA is undefined
> if double(!version.release) GE 5.4 then \$
> extra = {utc: 1}
> value = systime(_EXTRA=extra)
>
> Anyway you cut it, you will need an IDL version check.

I think this might depend on how old a version of IDL
you were calling. I don't think _Extra always failed

gracefully in the early days, when routines were passed a keyword they didn't understand.

On the other hand, this does have the advantage of running in a run-time version of IDL, whereas the CALL_FUNCTION approach would fail.

Cheers,

David

--

David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Unsupported keyword on older IDL version
Posted by [JD Smith](#) on Mon, 29 Jul 2002 20:17:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Mon, 29 Jul 2002 08:14:12 -0700, Craig Markwardt wrote:

> Paul Hick <pphick@ucsd.edu> writes:
>
>> This kind of problem must have come up in this group before, but here
>> it is anyways.
>>
>> If I call the IDL systemtime function with the utc keyword:
>> print, systemtime(/utc)
>>
>> on IDL versions older than 5.4 I get the expected 'keyword not
>> supported' error message, since, well, it's not supported, right? The
>> error message pops up when the function containing the systemtime call is
>> compiled, so testing e.g. !version.release before executing the systemtime
>> call doesn't work.
>>
>> The only solution I have come up with so far is to put the systemtime call
>> inside a call_function call:
>> print, call_function('systemtime', /utc)
>> This will compile (I will still need to test for !version.release
>> before actually executing the statement of course).
>
> Another possibility is to use the _EXTRA mechanism:
>

```
> ;; Assume EXTRA is undefined
> if double(!version.release) GE 5.4 then $
>   extra = {utc: 1}
>   value = systime(_EXTRA=extra)
>
> Anyway you cut it, you will need an IDL version check.
```

Unless you use `_STRICT_EXTRA`, the inheritance mechanism doesn't care if you pass it bogus keywords... in fact, that deficiency is what prompted the creation of `_STRICT_EXTRA`. Here you can use it to your advantage.

```
IDL> print,systime(/UTCABC)
% Keyword UTCABC not allowed in call to: SYSTIME % Execution halted at:
$MAIN$
IDL> print,systime(_EXTRA={UTCABC:1})
Mon Jul 29 13:08:09 2002
IDL> print,systime(_STRICT_EXTRA={UTCABC:1}) % Keyword UTCABC not allowed
in call to: SYSTIME % Execution halted at: $MAIN$
```

Unfortunately, for this case, it doesn't work, because `SYSTIME` went from having none to some keywords, so in old versions of IDL (prior to the introduction of keywords to `SYSTIME`), you get:

```
IDL> print,systime(_EXTRA={UTC:1})

print,systime(_EXTRA={UTC:1})
      ^
% Keyword parameters not allowed in call.
```

For routines with new keywords, which always had keywords, this should work.

JD

Subject: Re: Unsupported keyword on older IDL version
Posted by [David Fanning](#) on Mon, 29 Jul 2002 20:35:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

I wrote earlier today:

```
> On the other hand, this does have the advantage of
> running in a run-time version of IDL, whereas the
> CALL_FUNCTION approach would fail.
```

I was apparently wrong about this. Dick Jackson points out to me in a private e-mail that he can include `EXECUTE` and `CALL_FUNCTION` in run-time programs and they

work perfectly OK.

I'm embarrassed to say I didn't try this myself when I wrote this advice, but based it on a recent posting to this newsgroup from the usually reliable Reimar Bauer, who wrote:

```
"there is a small amount of rules you should know.  
[about run-time IDL]  
* !PATH does not exist in runtime  
* the filename of the routine have to be the same as  
the calling procedure.  
* EXECUTE won't work (e.g. read_ascii uses EXECUTE)  
* nothing could be compiled but other compiled routines  
could be loaded."
```

Here is a test program. Runs fine on Windows 2000 with IDL 5.5 in run-time. Does anyone else have problems with it?

```
*****  
PRO RuntimeFeaturesTest  
READ_JPEG, FILEPATH('rose.jpg', SUBDIR=['examples', 'data']), image  
IF image[0] NE -1 THEN TV, /True, image  
Wait, 1  
Call_Procedure, 'Plot', RandomU(seed, 10)  
Wait, 1  
pixels = TVRd()  
count = Call_Function('Total', pixels)  
XYOutS, /Device, 100, 100, 'Total of pixels: '+StrTrim(count, 2)  
Wait, 1  
ok = Execute('Erase')  
Wait, 1  
END  
*****
```

```
IDL> runtimefeaturestest  
IDL> Resolve_All  
IDL> Save, File='runtimefeaturestest.sav', /Routines
```

What do you make of this, Reimar?

Cheers,

David

--
David W. Fanning, Ph.D.

Subject: Re: Unsupported keyword on older IDL version
Posted by [Paul Hick](#) on Tue, 30 Jul 2002 02:41:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

Craig Markwardt wrote:

```
> Paul Hick <pphick@ucsd.edu> writes:
>
>
>> This kind of problem must have come up in this group before, but here it
>> is anyways.
>>
>> If I call the IDL systime function with the utc keyword:
>> print, systime(/utc)
>>
>> on IDL versions older than 5.4 I get the expected 'keyword not
>> supported' error message, since, well, it's not supported, right? The
>> error message pops up when the function containing the systime call is
>> compiled, so testing e.g. !version.release before executing the systime
>> call doesn't work.
>>
>> The only solution I have come up with so far is to put the systime call
>> inside a call_function call:
>> print, call_function( 'systime', /utc)
>> This will compile (I will still need to test for !version.release before
>> actually executing the statement of course).
>>
>
> Another possibility is to use the _EXTRA mechanism:
>
> ;; Assume EXTRA is undefined
> if double(!version.release) GE 5.4 then $
>   extra = {utc: 1}
>   value = systime(_EXTRA=extra)
>
> Anyway you cut it, you will need an IDL version check.
>
> Craig
>
>
```

The `_extra` mechanism is a fairly recent addition to IDL, isn't it? So

that would just move the problem to a different arena.

The systime function is 'special' in the sense that it is a 'built-in' IDL function (there is no systime.pro file floating around anywhere that I can see). That suggests another solution. The systime(/utc) call could be hidden in a separate user-written function systime_for_idl5_4. Calling this after checking !version.release would also work, since the user-written routine would never get compiled on pre5.4 versions.

Paul

Subject: Re: Unsupported keyword on older IDL version
Posted by [David Fanning](#) on Tue, 30 Jul 2002 03:21:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paul Hick (pphick@ucsd.edu) writes:

> The _extra mechanism is a fairly recent addition to IDL, isn't it?

Yes, I think it has only been added in the past 10 or 12 years.

Cheers,

David

P.S. Let's just say the response would probably be overwhelming if we started an IDL 4.0 branch of the IDL Expert Programmer's Association. :-)

--

David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Unsupported keyword on older IDL version
Posted by [Paul Hick](#) on Tue, 30 Jul 2002 03:30:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

William Thompson wrote:

> Paul Hick <pphick@ucsd.edu> writes:

>

>

>> This kind of problem must have come up in this group before, but here it
>> is anyways.
>>
>
>> If I call the IDL systime function with the utc keyword:
>> print, systime(/utc)
>>
>
>> on IDL versions older than 5.4 I get the expected 'keyword not
>> supported' error message, since, well, it's not supported, right? The
>> error message pops up when the function containing the systime call is
>> compiled, so testing e.g. !version.release before executing the systime
>> call doesn't work.
>>
>
>> The only solution I have come up with so far is to put the systime call
>> inside a call_function call:
>> print, call_function('systime', /utc)
>> This will compile (I will still need to test for !version.release before
>> actually executing the statement of course).
>>
>
>> This to me looks like a somewhat illegitimate use of call_function. Not
>> that I mind bending the rules a bit, but are there any other (better)
>> techniques of dealing with keywords that don't work in all IDL versions.
>>
>
>
> Why do you call this "illegitimate"? To me this seems like just the sort of
> thing that call_function() was designed for. I've always done it this way in
> the past. In fact, I've usually used execute() in the past, but
> call_function() is more elegant.
>
> Bill Thompson
>

My main use for call_function has been as the IDL version of the Fortran
'external' mechanism: passing function names as arguments to
procedures that would then call the function using call_function. That's
a lot more powerful than the current keyword problem. But of course you
are right: whatever works

Paul
