
Subject: Re: Return UNDEFINED from a function
Posted by [David Fanning](#) on Wed, 11 Oct 2006 23:04:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ed Hyer writes:

> Shouldn't the RETURN statement check to see that its argument is
> defined? Am I wrong that this is bad compiler behavior?

An undefined variable is a perfectly valid variable type
in IDL. Why wouldn't you be able to return it? :-)

OK, maybe it's weird. But I don't think it's necessarily
"bad". I've even used the feature on occasion.

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: Return UNDEFINED from a function
Posted by [Mariolncandenza](#) on Wed, 11 Oct 2006 23:09:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

BTW, this behavior is NOT dependent on setting the ON_ERROR. Same
results using ON_ERROR,0

Subject: Re: Return UNDEFINED from a function
Posted by [Mariolncandenza](#) on Wed, 11 Oct 2006 23:28:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

BTW, this behavior is NOT dependent on setting the ON_ERROR. Same
results using ON_ERROR,0

Subject: Re: Return UNDEFINED from a function
Posted by [Mariolncandenza](#) on Thu, 12 Oct 2006 08:04:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

> OK, maybe it's weird. But I don't think it's necessarily
> "bad". I've even used the feature on occasion.

I fail to see how this "feature" could be used. If you call a function as
value = FUNCTION(args)
and FUNCTION() returns an <UNDEFINED>, this will cause an error.

Of course, I'm not using my imagination enough. If you have a procedure, and you're not feeling like using any normal error-handling mechanism, you can simply define it as a function and execute it using HELP, and use a HELP-parsing routine to determine if it ran correctly or not. Is there an obfuscated IDL contest?

Subject: Re: Return UNDEFINED from a function
Posted by [JD Smith](#) on Thu, 12 Oct 2006 17:53:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thu, 12 Oct 2006 01:04:44 -0700, Ed Hyer wrote:

>> [quoted text muted]
>
> I fail to see how this "feature" could be used. If you call a function as
> value = FUNCTION(args)
> and FUNCTION() returns an <UNDEFINED>, this will cause an error.
>
> Of course, I'm not using my imagination enough. If you have a procedure,
> and you're not feeling like using any normal error-handling mechanism, you
> can simply define it as a function and execute it using HELP, and use a
> HELP-parsing routine to determine if it ran correctly or not. Is there an
> obfuscated IDL contest?

I think it's a shame IDL doesn't have undef() for creating a real, undefined anonymous variable (you can fake it just by mentioning an undeclared variable). And, analogously, defined() to test if a variable is defined (which reads better than 'n_elements(var) ne 0'). But in any case, you could always have something like:

```
if n_elements(function(args,OTHER=other)) ne 0 then do_something_with,other
```

Not exactly a standard paradigm, but it might come in useful somewhere. In particular, if FUNCTION passes an argument through, and that argument is undefined, it would be nice to keep it undefined.

JD

Subject: Re: Return UNDEFINED from a function

Posted by [Jean H.](#) on Thu, 12 Oct 2006 18:21:47 GMT

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

JD Smith wrote:

> On Thu, 12 Oct 2006 01:04:44 -0700, Ed Hyer wrote:

>

>

>>> [quoted text muted]

>>

>> I fail to see how this "feature" could be used. If you call a function as

>> value = FUNCTION(args)

>> and FUNCTION() returns an <UNDEFINED>, this will cause an error.

>>

>> Of course, I'm not using my imagination enough. If you have a procedure,

>> and you're not feeling like using any normal error-handling mechanism, you

>> can simply define it as a function and execute it using HELP, and use a

>> HELP-parsing routine to determine if it ran correctly or not. Is there an

>> obfuscated IDL contest?

>

>

> I think it's a shame IDL doesn't have undef() for creating a real,

> undefined anonymous variable (you can fake it just by mentioning an

> undeclared variable). And, analogously, defined() to test if a

> variable is defined (which reads better than 'n_elements(var) ne 0').

> But in any case, you could always have something like:

>

> if n_elements(function(args,OTHER=other)) ne 0 then do_something_with,other

>

> Not exactly a standard paradigm, but it might come in useful

> somewhere. In particular, if FUNCTION passes an argument through, and

> that argument is undefined, it would be nice to keep it undefined.

>

> JD

In this case, what would be the interest in keeping a function rather than a procedure? ... the result of the function, if defined, is of prime interest... with this method one would have to run the function a 2nd time in order to get the result value.

I guess one could do
proName,args, result

If args are defined, n_elements(results) will be > 0 (with all the proper checks in the procedure, so ****nothing**** is done if args are missing.

Jean

Subject: Re: Return UNDEFINED from a function
Posted by [JD Smith](#) on Thu, 12 Oct 2006 19:54:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thu, 12 Oct 2006 12:21:47 -0600, Jean H. wrote:

- > In this case, what would be the interest in keeping a function rather
- > than a procedure? ... the result of the function, if defined, is of
- > prime interest... with this method one would have to run the function a
- > 2nd time in order to get the result value.
- >
- > I guess one could do
- > proName,args, result
- >
- > If args are defined, n_elements(results) will be > 0 (with all the
- > proper checks in the procedure, so **nothing** is done if args are missing.

I admit it's not the most useful thing (it would be much more useful if UNDEFINED were a valid variable type, which could be assigned to other variables), but that doesn't mean there's no conceivable better use.

Subject: Re: Return UNDEFINED from a function
Posted by [badjelly.witch](#) on Thu, 12 Oct 2006 21:27:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

- > An undefined variable is a perfectly valid variable type
- > in IDL. Why wouldn't you be able to return it? :-)
- >
- > OK, maybe it's weird. But I don't think it's necessarily
- > "bad". I've even used the feature on occasion.

Spoken like a true IEP!

For what purpose have you used this feature, David?

Subject: Re: Return UNDEFINED from a function
Posted by [David Fanning](#) on Fri, 13 Oct 2006 02:52:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mark Hadfield writes:

- > Spoken like a true IEP!
- >
- > For what purpose have you used this feature, David?

I use it with text widgets when I want to know if the user actually typed something into the text widget. Maybe they cleared the text that was there, but didn't type anything else. This is hard to check for because the text isn't really **there**, if you know what I mean.

Here, for example, is code from FSC_INPUTFIELD:

```
IF String(testValue) EQ "NULLVALUE" THEN BEGIN
  Ptr_Free, self.theValue
  self.theValue = Ptr_New(/Allocate_Heap)
ENDIF ELSE *self.theValue = testValue
```

I return self.theValue, which I guess technically is NOT an undefined variable, but is a pointer to an undefined variable. But I can test it to see if it is undefined.

Humm. Maybe this isn't **exactly** what Ed was complaining about, but I still don't think it is a bug. :-)

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.")
