
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
