
Subject: Error Handling Change in IDL 8
Posted by [David Fanning](#) on Tue, 19 Oct 2010 15:16:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Folks,

This change to ON_ERROR,2 error handling is driving me nuts!!

I've very close to just giving up on IDL 8. I do NOT want to re-write the last 10 years of my IDL programs! :-(

Anyone else having this problem?

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: Error Handling Change in IDL 8
Posted by [Paul Van Delst\[1\]](#) on Tue, 19 Oct 2010 16:35:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> Folks,
>
> This change to ON_ERROR,2 error handling is driving me nuts!!

Sorry, man.

> I've very close to just giving up on IDL 8. I do NOT want
> to re-write the last 10 years of my IDL programs! :-(

Well, not all at once. To quote James Coplien in the foreword to "Clean Code":

<quote>

...the bulk of the work lies not in manufacturing but in maintenance -- or its avoidance. In software, 80% or more of

what we do is quaintly called "maintenance": the act of repair.

</quote>

Granted this is indirect repair, but it is repair nonetheless.

Think of it as fixing your house after a storm blows through, as opposed to fixing your house after you put the ladder through the living room window (and then proceed to knock over your antique Tiffany lamp from the table....)

In this case ITTVIS is the "Act of God" causing the damage. :o)

> Anyone else having this problem?

No, I use CATCH.

I will offer more than just a sympathetic ear, though.

Can you replace that one line of code

ON_ERROR, 2

in your routines with a CATCH construct that replicates the behaviour you want?

If so (I don't think there should be a difference between functions and procedures in this case, but ?), put that code

in an include file and simply replace all instances of

ON_ERROR, 2

with

@<your catch construct include file>

You can trivially do this replacement with a script (see a previous post of mine about this).

FWIW, if you come up with the include file and create a branch for me in your subversion repository, I can check that

branch out, do the replacements, and commit the changes (I will need write access so I

understand if that makes you feel

a bit squirrely and say no). You can then test the changes in the branch (I can also if you have standard tests for the

code). If you like what you see, we can merge the branch into the trunk. If you don't like it, we can simply delete the

branch.

cheers,

paulv

Subject: Re: Error Handling Change in IDL 8

Posted by [David Fanning](#) on Tue, 19 Oct 2010 17:27:34 GMT

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

Paul van Delst writes:

> Sorry, man.

> Think of it as fixing your house after a storm blows through, as opposed to fixing your house after you put the ladder
> through the living room window (and then proceed to knock over your antique Tiffany lamp from the table....)

This sounds suspiciously like the advice Coyote often gives me. I don't like this any better than I like his. :-(

> Can you replace that one line of code
> `ON_ERROR, 2`
> in your routines with a CATCH construct that replicates the behaviour you want?

Sigh...Yes, I can replace them with CATCH, which is what I am doing as I find them.

> If so (I don't think there should be a difference between functions and procedures in this case, but ?), put that code
> in an include file and simply replace all instances of
> `ON_ERROR, 2`
> with
> `@<your catch construct include file>`
>
> You can trivially do this replacement with a script (see a previous post of mine about this).

Well, in functions the CATCH returns some value.
In procedures, it doesn't. Most of these `ON_ERROR,2` calls (but not all) are in functions, and I try to be a bit careful about what I return. I'm not sold on the notion of a generic CATCH handler.

> FWIW, if you come up with the include file and create a branch for me in your subversion repository, I can check that
> branch out, do the replacements, and commit the changes (I will need write access so I understand if that makes you feel
> a bit squirrely and say no). You can then test the changes in the branch (I can also if you have standard tests for the
> code). If you like what you see, we can merge the branch into the trunk. If you don't like it, we can simply delete the
> branch.

It is not really my library routines I am worrying about. I'm pretty careful with Library routines. I'm less careful when writing one-offs to do science. It's these routines I am struggling with at the moment. It's just disconcerting to get error messages that don't mean a damn thing to me and do nothing to help me solve the problem.

Alright, I'm going to quit whining about it. But I still think changing the way error handling works was a terrible idea and almost guaranteed to antagonize long-time customers. (If I'm the only one antagonized, as it appears, then perhaps Coyote is right that "Nobody uses damn error handling anyway!")

Cheers,

David

P.S. I'm not sure how to use a Ruby script. Do you have that Perl script around? I can think of a few ways to use that! Thanks.

--

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: Error Handling Change in IDL 8
Posted by [Paul Van Delst\[1\]](#) on Tue, 19 Oct 2010 18:31:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

- > Well, in functions the CATCH returns some value.
- > In procedures, it doesn't. Most of these ON_ERROR,2 calls
- > (but not all) are in functions, and I try to be a bit
- > careful about what I return. I'm not sold on the notion
- > of a generic CATCH handler.

Yeah, it's not straightforward I agree.

For what I do, I'm thinking about adopting
RETURN, !NULL
for my "generic" function error handler.

I've only thought about it for about 5 minutes, so there may be unforeseen pitfalls to this approach, but I prefer convention over configuration.

- > It is not really my library routines I am worrying
- > about. I'm pretty careful with Library routines.

- > I'm less careful when writing one-offs to
- > do science. It's these routines I am struggling with
- > at the moment. It's just disconcerting to get error
- > messages that don't mean a damn thing to me and do
- > nothing to help me solve the problem.

I'm sorry.... "one-offs"? I associate that term with throwaway code. So, two things immediately spring to mind:

- 1) does anyone really write "one-offs" anymore? :o)
- 2) If they truly are "one-offs", what the hell are you doing worrying about error handling? Just use `ON_ERROR, 0` !!

:o)

- > Alright, I'm going to quit whining about it. But
- > I still think changing the way error handling works
- > was a terrible idea and almost guaranteed to
- > antagonize long-time customers. (If I'm the only
- > one antagonized, as it appears, then perhaps
- > Coyote is right that "Nobody uses damn error
- > handling anyway!")

I don't think we should stop whining about it. I'm usually the first in line to whinge about decisions made by ITTVIS

but in this instance, it didn't affect me at all since I use `CATCH` exclusively. (Similarly for the meaning of "-1" being returned by an unsuccessful `WHERE`, etc etc)

- > P.S. I'm not sure how to use a Ruby script. Do you have
- > that Perl script around? I can think of a few ways to use
- > that! Thanks.

Well, for ruby, on your *nix system type

```
$ irb
```

If you see something like

```
irb(main):001:0>
```

then you have successfully entered the interactive ruby shell and it means ruby is installed on your system. To run a

ruby script, you should be able to just type

```
$ ruby script_name.rb
```

to run it, but there may be some envvars that need setting... can't recall.

But, regardless, I will ask my colleague if he has his perl script lying around (since I can't find it in our svn repo).

cheers,

paulv

Subject: Re: Error Handling Change in IDL 8
Posted by [David Fanning](#) on Tue, 19 Oct 2010 18:51:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paul van Delst writes:

> I'm sorry.... "one-offs"? I associate that term with throwaway code.
So, two things immediately spring to mind:
> 1) does anyone really write "one-offs" anymore? :o)
> 2) If they truly are "one-offs", what the hell are you doing worrying about error handling? Just use
> ON_ERROR, 0 !!

Well, because of how much pressure I was feeling to produce a result TODAY, I sometimes wrote what I thought were one-offs. After that bit me in the rear about a dozen times, I didn't care HOW LONG it took to write the code, it would be reusable. I got a reputation as a slow programmer (at least the first time they wanted something done). After that, I was MUCH faster. Reminds me a bit of IDL, in that respect. :-)

I guess what I really mean is "casual code", code that is not really meant to be Library code, but code that is useful to me, nonetheless. Maybe it's code where I am exploring some kind of idea or approach to the problem. I'm not really interested in typing Catch paragraphs, but I do what to get meaningful error messages.

> But, regardless, I will ask my colleague if he has his perl script
> lying around (since I can't find it in our svn repo).

Appreciate it!

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: Error Handling Change in IDL 8

Posted by [penteado](#) on Tue, 19 Oct 2010 19:09:16 GMT

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

On Oct 19, 3:27 pm, David Fanning <n...@dfanning.com> wrote:

>
> Alright, I'm going to quit whining about it. But
> I still think changing the way error handling works
> was a terrible idea and almost guaranteed to
> antagonize long-time customers. (If I'm the only
> one antagonized, as it appears, then perhaps
> Coyote is right that "Nobody uses damn error
> handling anyway!")

I do not feel strongly about this, but it is looking like it would be better to either revert to the old behavior in the next release, or make it a preference. I find it better to use a preference, that way everybody can have their preferred behavior. The same way that in a static compiled language there is the option to tell the compiler to add traceback information and runtime checks.

Subject: Re: Error Handling Change in IDL 8

Posted by [penteado](#) on Tue, 19 Oct 2010 19:13:32 GMT

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

On Oct 19, 5:09 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:

> I do not feel strongly about this, but it is looking like it would be
> better to either revert to the old behavior in the next release, or
> make it a preference. I find it better to use a preference, that way
> everybody can have their preferred behavior. The same way that in a
> static compiled language there is the option to tell the compiler to
> add traceback information and runtime checks.

Though it may only take a few more bugs in Graphics where "on_error,2" was used to make me feel strongly in favor of the old behavior. That last one (with window()'s title) was found and dealt with, despite the troubles with on_error and setting breakpoints.

Subject: Re: Error Handling Change in IDL 8

Posted by on Wed, 20 Oct 2010 09:58:28 GMT

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

On Oct 19, 9:13 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:

> On Oct 19, 5:09 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
>
>> I do not feel strongly about this, but it is looking like it would be
>> better to either revert to the old behavior in the next release, or

>> make it a preference. I find it better to use a preference, that way
>> everybody can have their preferred behavior. The same way that in a
>> static compiled language there is the option to tell the compiler to
>> add traceback information and runtime checks.
>
> Though it may only take a few more bugs in Graphics where "on_error,2"
> was used to make me feel strongly in favor of the old behavior. That
> last one (with window()'s title) was found and dealt with, despite the
> troubles with on_error and setting breakpoints.

Maybe a guide from ITTVIS on IDL error handling would be helpful in many situations. Is this available?
