

Re: False exceptions?" (was Re: theme of the week: tools

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2004-09/4974.html>

From: Dan Perl (danperl_at_rogers.com)

Date: 09/28/04

Date: Tue, 28 Sep 2004 02:11:38 -0400

Stephan,

Look at it from my point of view. First script I debug with Wing IDE I get one of those false positives. I am debugging a script and debugging includes looking for exceptions and this debugging tool tells me to ignore exceptions. It's annoying and I'm trying to figure it out and I see an explanation that sounds like "this is not really a problem and it is happening because Wing IDE is so great". What would you be thinking?

I do believe now that from a technical point of view you made the right decision. That's because you are giving users both choices (there is actually also a third choice, "Always Stop"). I figured out only after trying Wing that it was possible to run also in a mode where the debugger does not stop at the point where the exception is raised. So you are giving both modes, one that is very powerful (even if it has the "false positives") and one that is good enough in most cases, IMO, and does not have the "false positives". That's all fair.

However, I still think that the "official" explanation for the "false positives" is pathetic. I didn't see it in only one place, it's all over. It was the explanation I first got in the IDE when I encountered the "false positive" and I found it throughout the documentation and in support replies to users (it's amazing what one can find). Basically, the explanation is "it's not a bug, it's a feature". And "just ignore". And it will happen only 1-2 times, or 3-4 times...

Here's how I would have explained it. "This exception may be a false positive. We are sorry about that, but here is why we are doing it this way, and we are also giving you the option to run it the other way." Your posting was actually in line with this. You even used that valuable "sorry" word. Thank you.

Obviously, I was more sensitive to it than other people. Why? I don't know. Maybe it just caught me at a bad time, but I still think it's wrong. BTW, the "useful exception related feature" makes me think of Bush's "weapons of mass destruction-related program activities". That's how low I

think of it. (I know you're not responsible for that, Stephan, you didn't have to apologize for it.)

As for my choice? I will stick with Komodo. Not only because I already paid for it, but I think it does fit my needs better. I am using it for personal use, on a one-person project of my own. Something like Wing's exception detection or other power tools are not really useful for me. I just started using wxPython and maybe as I get deeper into that I will be proven wrong, but exceptions are normally not even a big issue for me. My debugging is usually only for correctness of results, so breakpoints and stepping through is all I need. For the rare exception that I can't figure out from the traceback, I can always just put a breakpoint and run again. I like the UI of Komodo more (sorry, Stephan, but Wing's Project view sucks, I hope you didn't work on that too). I even prefer the code completion the way it's done in Komodo, but maybe that's because I got used to it. And once again, a class browser is important to me. Wing's drop-down selections for classes and methods don't make up for it, they don't give that graphical view of a class's structure. In a nutshell, I used Komodo for a month and I was absolutely satisfied with it; I still thought I should make an informed decision so I tried also Wing, but after 2 days I had only disappointments.

Dan

"Stephan Deibel" <sdeibel@wingware.com> wrote in message
news:mailman.3990.1096338692.5135.python-list@python.org...

> Hi,

>

> *I'm the dope that co-wrote this thing, so here's the technical background:*

>

> *The "false exceptions" thing is a technical limitation of detecting
> whether or not an exception is going to lead to program termination at the
> moment it is raised, rather than later when exiting the program. We go up
> the stack to inspect Python byte code, and since we can't see into C/C++
> object code we sometimes get it wrong. We call it a "false positive"
> because it's the same as a blood test telling you you've got something
> that you don't.*

>

> *Believe it or not, we were not just being dumb: We weighed having a
> necessarily imperfect but useful feature with not having the feature at
> all. Same as having an imperfect blood test rather than not at all. In
> fact, from feedback, it seems most people agree that ignoring a few
> exceptions during the first debug run is worth always being able to
> inspect the unaltered program state seen immediately at the moment the
> exception is raised (before, e.g., 'finally' clauses are executed).
> That's why it's on by default, but can be turned off in prefs.*

>

> *Maybe we got that wrong, but I dunno... the people we hear from may not
> represent the overall experience.*

>

> *Dan Perl wrote:*

>> *No, this is NOT a "useful exception related feature", it's a workaround*

comp.lang.python: Re: False exceptions?" (was Re: theme of the week: tools

>> *for a bug.*

>

> *I think I'd have to agree that the paragraph you cited muddles up the
> utility of ignoring exceptions with under-explaining the false positives.*

> *The feature is not just there as a workaround but it's probably fair to
> say that's 99% of what it's used for. Sorry about that.*

>

> *BTW, the 10 day trial can be renewed automatically. Then if you run out,
> just ask for more time. No problem. Deciding the duration of trials is a
> nasty business decision that's not easy to make. I can't really make
> excuses -- it's just how it is right now.*

>

> *Stephan Deibel*

>

> --

> *Wingware*

> *Wing IDE for Python*

> *Advancing Software Development*

>

> *www.wingware.com*