

Re: Where to start the try block and what's the overhead???

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2004-05/3146.html>

From: Alvin Bruney [MVP] (*vapor*)

Date: 05/14/04

Date: Fri, 14 May 2004 11:11:03 -0500

try blocks add overhead when they are fired or spring into action. Otherwise they don't. The issue with wrapping a big nasty try block is that you aren't really able to tell what entity fired the exception.

The implicit assumption in a try block is that you are prepared to handle a particular type of exception. A big all encompassing try block defeats that purpose since your code cannot possibly be prepared to handle any and all exceptions. In fact, I am a proponent of letting certain exceptions blow the application apart – a stack overflow or memory exhaustion exception are good examples of that.

--

Regards,

Alvin Bruney

[ASP.NET MVP <http://mvp.support.microsoft.com/default.aspx>]

Got tidbits? Get it here... <http://tinyurl.com/27cok>

"Bob Rock" <nospam.yet_another_apprentice@hotmail.com> wrote in message news:%23iC3jvcOEHA.3596@tk2msftngpl3.phx.gbl...

> Hello,

>

> this is something I've been asking myself for sometime ... and now I'd

> like

> to clarify this point.

> When should the try block start in a method??? What I mean is, does having

> all the code instead of a smaller set of it inside a try clause add any

> overhead??? 1What I'd like to understand is if, to be completely sure that

> no unhandles exception will get to the user, I can place all the code

> inside

> a try block

> and if this practice adds unnecessary overhead (memory usage, more cpu

> cycles, etc.) or if having a smaller set of instructions instead of all

> the

> code under a try has the EXACT SAME effect on resources (overhead).

>

>

> Bob Rock

>

>

>