

Re: Handling an exception

Source:

http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.distributed_apps/2006-02/msg00045.html

- *From:* "Ross" <Ross@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 20 Feb 2006 08:36:12 -0800
-

Thanks Jay

Appreciate your help

Kind regards

Ross Petersen

"Jay B. Harlow [MVP - Outlook]" wrote:

Ross,
| to start SQL server & connect to the DB then if it s successful then have
a
| line saying Resume Next & then continue with whatever I wanted to do.

The "best" you could do is to put a Try/Catch around each of the commands you want retry and have the Catch block retry the Try Block. I find using Goto in the Catch block the "easiest" way to "Retry", others have put the entire Try/Catch in a loop...

BTW: I've heard all the arguments about how Goto is evil & should be avoided, in this case the "Goto Retry" is more like a "Retry" statement. Yes "goto retry" could be used for evil, however it can also be used for good...

Something like:

Try

Retry:

'...something

Catch ex as FileNotFoundException

Re: Handling an exception

```
If MessageBox.Show("File does not exit!", _  
Application.ProductName, _  
MessageBoxButtons.RetryCancel, _  
MessageBoxIcon.Question, _  
MessageBoxDefaultButton.Button2) _  
= DialogResult.Retry Then  
GoTo Retry  
End If
```

End Try

Caution: With either the Goto Retry or a loop, be certain to allow your users an Out, so you don't get into an endless loop.

Hope this helps
Jay

--

Hope this helps
Jay [MVP - Outlook]
..NET Application Architect, Enthusiast, & Evangelist
T.S. Bradley - <http://www.tsbradley.net>

```
"Ross" <Ross@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message  
news:71D32336-0052-4BAE-A6FB-6C9B2B8FE93A@xxxxxxxxxxxxxxxxxxxx  
| Hi folks  
|  
| I was wondering, suppose I had some code that went like this:  
|  
| Try  
|  
| cn = New SqlConnection  
| 'connection string to the db  
| strInfo = "server=(local);Persist Security  
| Info=False;pwd=;Integrated  
| Security=SSPI;database=FusionCommunityVolunteerService  
| ;Connect Timeout=30"  
|  
| 'open the connection  
| cn.ConnectionString = strInfo  
| cn.Open()  
|  
| Catch ex As SqlException  
|  
| Finally
```

Re: Handling an exception

```
|  
| If cn.State <> ConnectionState.Open Then  
| cn.Close()  
| cn.Dispose()  
| End If  
|  
| End Try  
|  
| Now, if I was using unstructured exception handling using the old fashioned  
| syntax (on error goto ...) & say SQL server wasnt running for some reason,  
| in  
| the error handler block, I could write some code that could try a few  
| times  
| to start SQL server & connect to the DB then if it s successful then have  
| a  
| line saying Resume Next & then continue with whatever I wanted to do.  
|  
| Now, with structured error handling as per the code sample above, how  
| would  
| I do that within the Catch ex as SqlException area. I know I can trap for  
| that exception, and I know I could have some code to try a few times to  
| get  
| SQL server started & connect to the DB, but suppose that is successful,  
| how  
| do I get the code to resume at the point where I do something once a  
| connection is successful. Do I have a line label at that point in the  
| code &  
| just have a line of code that says GoTo ...?  
|  
| It seems to me that that sort of technique is going back to the old  
| fashioned method and there must be a better way, or is there a facility  
| that  
| is the equivalent of Resume Next.  
|  
| Or perhaps .net will just automatically resume at the next lot of code if  
| you handle for a specific exception, although I dont think that is very  
| likely.  
|  
| Anyway, if somebody could give me a few pointers I would be most  
| appreciative.  
|  
| Kind regards  
|  
| Ross Petersen
```