

Re: Exception handling suggestions

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2007-03/msg01703.html>

- *From:* "Bruce Wood" <brucewood@xxxxxxxxxx>
 - *Date:* 11 Mar 2007 20:49:12 -0700
-

On Mar 11, 5:30 pm, "Zytan" <zytanlith...@xxxxxxxxxx> wrote:

```
main
{
    AppDomain.CurrentDomain.UnhandledException+=new
    UnhandledExceptionHandler(CurrentDomain_UnhandledException);
    Application.Run(new MyFirstForm());
}
```

Ok, if this is for a windows app, what is "main"? The main form's constructor? After InitializeComponent? No it can't be, because you are running the first form from this. Ah, it's in Program.cs! So, I have to change this file, and add that event handler addition line, I see.

You can add the event handler anywhere, but it's best to do it as early as possible so that the minimum amount of code runs before the handler is hooked up.

There is also the Application.ThreadException event. I always handle them both.

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfssystemwindowsformsapplicationclass>

Main, by the way, is the main entry point into any application, console or WinForms. It is usually the first piece of your code that the CLR calls, and it's usually where you set up global exception handlers.

Ok, Application.Run(new MyFirstForm()) does work here! It doesn't in a console app. It is really System.Windows.Forms.Application. I see.

```
private static void CurrentDomain_UnhandledException(object sender,
    UnhandledExceptionEventArgs e)
```

Re: Exception handling suggestions

```
{  
// The app shuts down..., you can log here..  
  
//After log exit  
Application.Exit();  
}
```

I presume this code must also go in Program.cs?

Well, not necessarily. For example, I have mine in a standard error logging class, and from my Main I call something like:

```
ErrorLog.Instance.SetupConsoleErrorHandling();
```

and inside there I subscribe to the events and the ErrorLog class contains the event handler methods. Nonetheless, that's just because I wanted a reusable utility. You can also have the method in your main program .cs file. Nothing wrong with that.

Finally, I would like to offer some general advice on catching exceptions: catch only what you can handle. Or, put another way, don't catch an exception unless you can take some substantial action regarding it. There are other threads in this newsgroup on exception handling... when and how to catch an exception, but I wanted to point that out because it explains why you (almost) never catch (Exception ex): because it's very rare that you can take some intelligent action with any exception that could possible happen.

The exception to that rule, of course, being logging, and that's what global exception handlers are for.

.