

Forum:[Hack Forums](#)

Thread: [\[Snippet\] Error Reporting System. \(Noob Friendly\)](#)

Exussit - May 10, 2013, 1:00 am

Ok, well I was bored and just tweaking with one of my applications, and thought it'd be a fun idea to create a little Error Reporting System for when shit goes wrong haha. Now this obviously isn't going to be the best example you could find on the internet I'm sure, but I hope it'll at least benefit some of you :)

Ok, so basically, the error reporting magic will generically begin in the catch region of your code if you're implementing try/catch/finally into it. (Which you really should be).

And it's as simple as this:

Define an integer in your class outside of a method (so you don't have to re-define it in every method of your class).

Like so:

Code `int k = 1;` Feel free to rename the integer whatever the hell you want, but for simplicity's sake, keep the value to 1.

Now here's the main method for it inclusive of the catch (obviously you'll need to have a try beforehand).

Code catch (Exception ex)

```
{
    string AppData = Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData);
    while (File.Exists(AppData + "/Exussit/Error Report No. " + k + ".txt"))
    {
        k++;
    }
    MessageBox.Show("An Error has occurred!" + Environment.NewLine + "Please Read:" +
Environment.NewLine + AppData + "/Exussit/Error Report No. " + k + ".txt" + Environment.NewLine
+ "for more details.", "Error!", MessageBoxButtons.OK, MessageBoxIcon.Error);
    File.WriteAllText(AppData + "/Exussit/Error Report No. " + k + ".txt", ex.ToString());
    Console.WriteLine("Created an Error Report in AppData.");
}
```

}Ok, I'll explain what this does bit by bit if you don't already know..

The GetFolderPath function basically retrieves (in this instance) the appdata folder because obviously, it's a unique directory on everyone's PC..

I've then used a while loop to say that whilst there is a file located in (Appdata Directory)/Exussit/ named Error Report No. "+k+".txt (i.e. the current value of the integer that we declared, so in this instance it's still "Error Report No. 1"), it will add +1 value to the integer that has been defined, and will keep on doing so until there is a file with a non-matching name, then proceed to create a new file name with the unallocated int.

The MessageBox line is rather self explanatory and provides information as to where the user can locate the created error report for handling.

For File.WriteAllText, this is the main part where it actually writes the error itself that has been caught (ex). It converts it to a readable string and then writes it to the designated text file.

The last line merely is for personal use in DebugTime and I've just used that to tell me the method has executed. (I prefer that to using breakpoints in some cases).

And well, that about wraps it up.

Now naturally, it's quite easy to add a little system on top of that which allows you to automatically send the report to a designated webhost or database; whatever you like if you're creative enough. I just thought I'd begin contributing a little more to the c# section considering it is my favourite programming language...

So anyway, hope this helped at least some of you haha. Enjoy :)

Alternative method: (Credit to Bausshacker)

If you prefer a more... uniformed approach to the file naming, you can remove the need for the int all together and go by an error type naming system instead.

Simply modify this line:

Code `File.WriteAllText(AppData + "/Exussit/Error Report No. " + k + ".txt", ex.ToString());` To this:
Code `File.AppendAllText(AppData + "/Exussit/Error Report. " + e + ".txt", ex.ToString());` And update the `MessageBox` function etc.. Correlating to the new 'e' value instead of the integer.

Happy programming!

BaussHacker - May 10, 2013, 3:40 am

Instead of giving the error files numeric names, why not sort them by the Exception Type?

Code `string ExceptionType = e.GetType().Name;`

Then use `File.AppendAllText()` rather than `File.WriteAllText()`

Much easier to look through your errors.

Exussit - May 10, 2013, 3:48 am

(May 10, 2013 03:40 AM)BaussHacker Wrote: [\[CITE\]](#) Instead of giving the error files numeric names, why not sort them by the Exception Type?

Code `string ExceptionType = e.GetType().Name;`

Then use `File.AppendAllText()` rather than `File.WriteAllText()`

Much easier to look through your errors. You're very clever, sir. This did not even cross my mind when writing it haha. It certainly would mitigate any tedium within larger projects... I'll add this as an alternative in the OP with credit to you :)