

Re: ApplicationException unhandled by user code

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2007-04/msg03297.html>

- *From:* Chris Benard <Chris@xxxxxxxxxxxxxxxx>
 - *Date:* 24 Apr 2007 05:52:23 -0700
-

Hi Barry.

If I understand what you're trying to do correctly, you want to catch any unhandled exceptions thrown by your application so that your program doesn't terminate or just so that you can handle the errors before termination.

If that is what you wish to do, you may want to look at:

`AppDomain.CurrentDomain.UnhandledException`

In this case, you'd change your line that reads:

```
Application.ThreadException += new ThreadExceptionHandler(new  
ThreadExceptionHandler().ApplicationThreadException);
```

to instead read:

```
AppDomain.CurrentDomain.UnhandledException += new  
ThreadExceptionHandler(new  
ThreadExceptionHandler().ApplicationThreadException);
```

Your signature for `ApplicationThreadException()` would change from:

```
public void ApplicationThreadException(object sender,  
ThreadExceptionEventArgs e)
```

to:

```
public void ApplicationThreadException(object sender,  
UnhandledExceptionEventArgs e)
```

That should let you catch all your exceptions, and in .Net 1.1, it will stop your program from exiting. In .Net 2.0, your program will still terminate on an unhandled exception, even if it is handled in this way. If you'd like the old .Net 1.1 behavior, put this in your

App.config:

```
<configuration>  
<runtime>  
<legacyUnhandledExceptionPolicy enabled="1" />  
</runtime>  
</configuration>
```

I hope this helps.

Chris Benard

Re: ApplicationException unhandled by user code

MCTS Windows, Web, SQL Server 2005

On Apr 24, 4:21 am, b...@xxxxxxxxxx wrote:

Hi,

I have the following Program.cs –

```
namespace TestFrameworkApplication
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.ThreadException += new
            ThreadExceptionHandler(new
            ThreadExceptionHandler().ApplicationThreadException);
            Application.Run(new FormMain());
        }

        /// <summary>
        /// Handles any thread exceptions
        /// </summary>
        public class ThreadExceptionHandler
        {
            public void ApplicationThreadException(object sender,
            ThreadExceptionEven
```