

MS Enterprise Library 5.0 (Logging Application Block)

Anubhav Tiwari

* R&D Dept., Syscom Corporation Ltd.

Abstract-

This paper talks about

- Why Microsoft Enterprise Library?
- How to use Logging Application Block - Microsoft Enterprise Library?
- What are advantages of using it?

Index Terms- Microsoft Enterprise Library, Logging Application Block, Types of Logging, How to Use, Advantages

I. INTRODUCTION

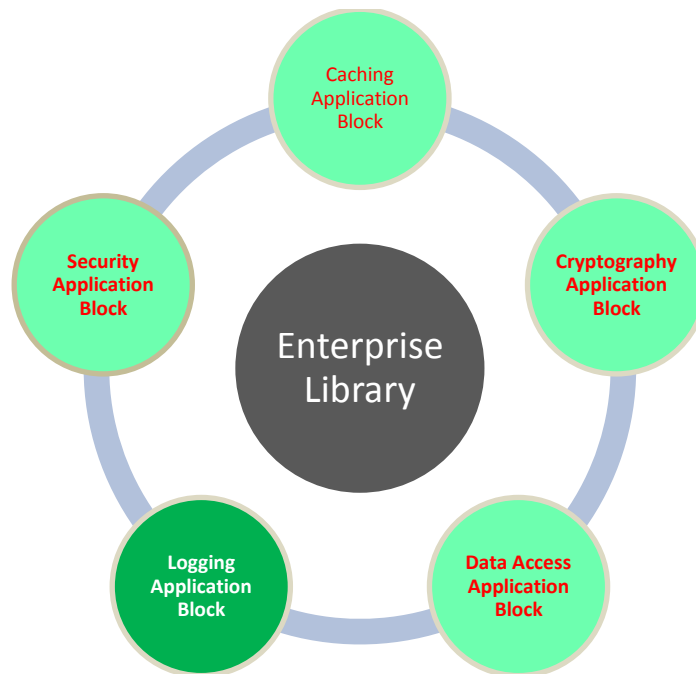
Dear Readers,

First of all, I would say this paper is just for ASP.net developers and I would like to start with a question.

“How many times you faced the problem where you were unable to find what the exact exception/Problem that occurred on production environment?”

Obviously, you would have faced the problem many-a-times and you must have spend considerable efforts to go to the root cause of the issue. But here is good news, Microsoft has provided “Enterprise Library” using which you can log the exceptions/errors.

Enterprise Library is a complete package for multiple blocks. Here I am not going with all the blocks as that will be too big a subject to digest at one go but obviously will tell you in details about Logging Application Block.



Types of Error Logging:

The Logging application block provides multiple ways to log the errors/exceptions of an application. These ways are as follows:

| S.No. | Ways of Error Logging |
|-------|-----------------------|
| 1 | Database Logging |
| 2 | Flat File Logging |
| 3 | Email Logging |
| 4 | Message Queue Logging |

Suitability of different versions:

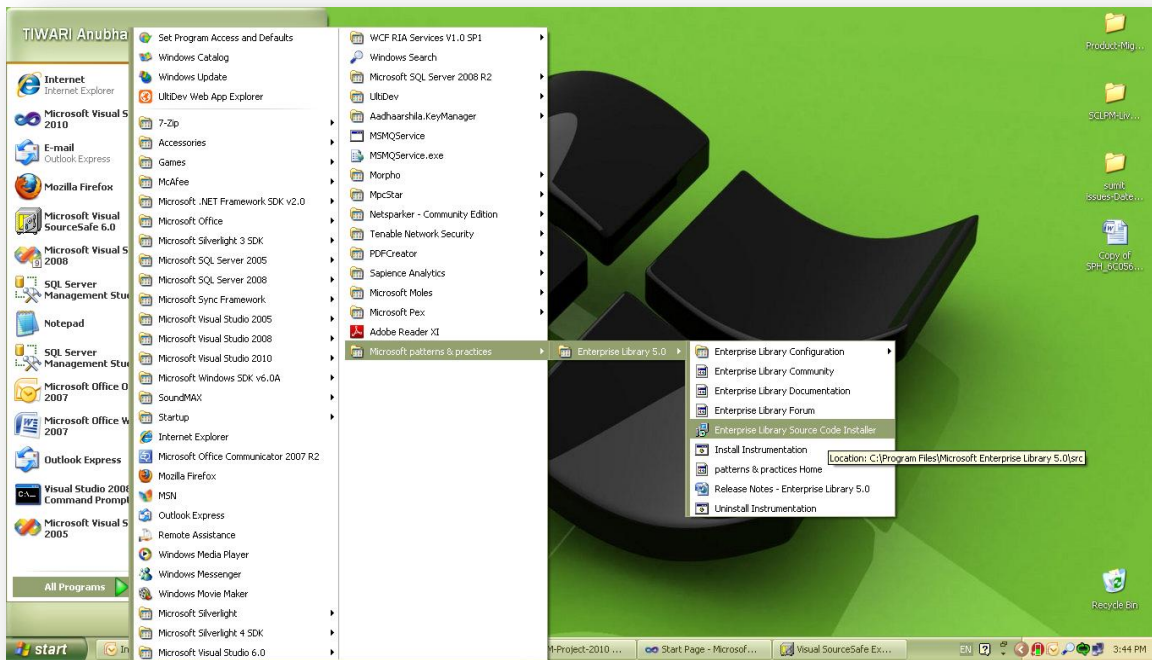
- 2.1 Version 2.0
 - a. Microsoft Enterprise Library 2.0 was released in January 2006. features include:
 - b. Supports .Net Framework 2.0
 - c. Significant improvements to Logging Application Block
- 2.2 Version 3.0
 - a. Microsoft Enterprise Library 3.0 was released in April 2007. New features include:
 - b. Supports .Net Framework 3.0
 - c. Contains Data Access, Cryptography, Exception Handling, Logging, Security Application Blocks
- 2.3 Version 4.0
 - a. Microsoft Enterprise Library 4.0 was released in May 2008. New features include:
 - b. Supports .Net Framework 3.5
 - c. Performance improvements (particularly, in the Logging Application Block)
 - d. Visual Studio 2008 support
- 2.4 Version 4.1
 - a. Microsoft Enterprise Library 4.1 was released in October 2008. New features include:
 - b. Supports .Net Framework 3.5
 - c. Performance improvements
 - d. Usability improvements to the Configuration Tool
 - e. Visual Studio 2008 Service Pack 1 support
- 2.5 Version 5.0
 - a. Microsoft Enterprise Library 5.0 was released in April 2010. New features include:
 - b. Supports .Net Framework 3.5 and above.
 - c. New Configuration Tool (Enterprise Library V5 Configuration tool)
 - d. Higher performing logging.
 - e. Visual Studio 2008 and Visual Studio 2010 support

How to use:

Now pick enterprise library for you and download it. For versions 5.0 download it from the below URL:
<http://www.microsoft.com/en-us/download/details.aspx?id=15104>

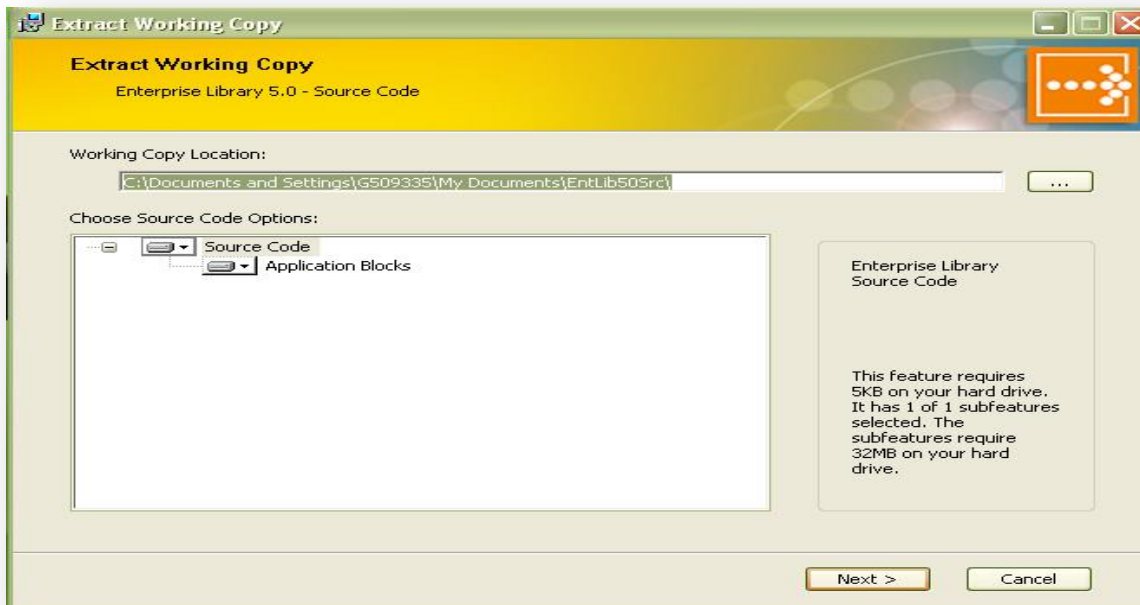


After installing Enterprise Library 5.0, we will have the installed program in our All Programs of windows as shown below in the screen shot.

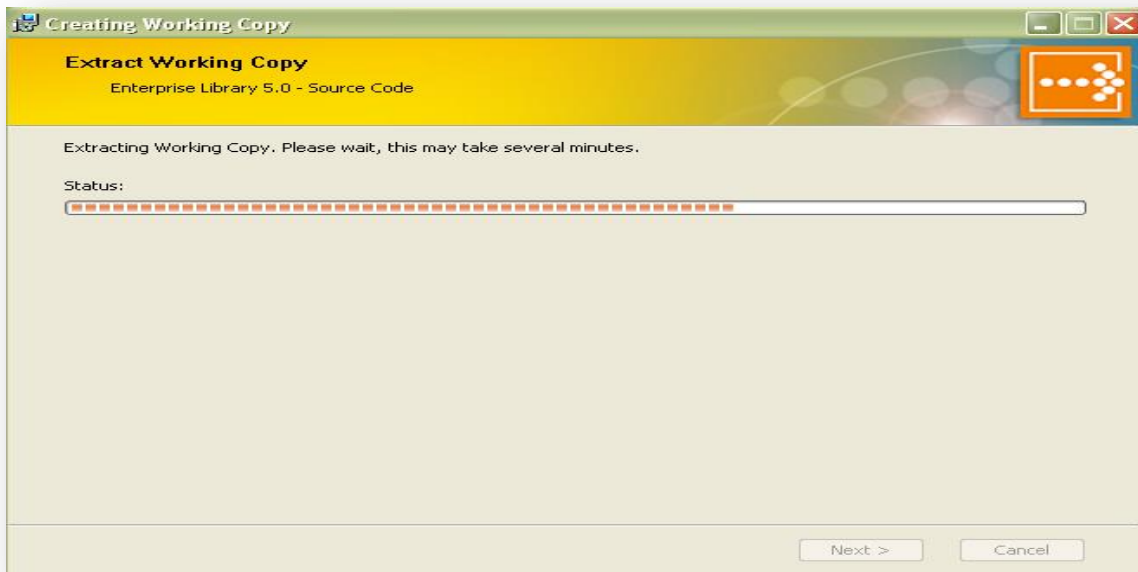


After the successful installation of Enterprise Library 5.0, we need to install the source code installer available in the All Programs. The source code installer provides us the source code needed to implement different application blocks of enterprise library. Here, in this white paper, we are concerned about implementation of Logging Application Block, so we will only look into the source of the corresponding block.

After the click on Enterprise Library Source Code installer, the following window appears as given in screen shot.



We can provide our custom path for unzipping the source code of the application block. The default location is My Documents >> EntLib50Src

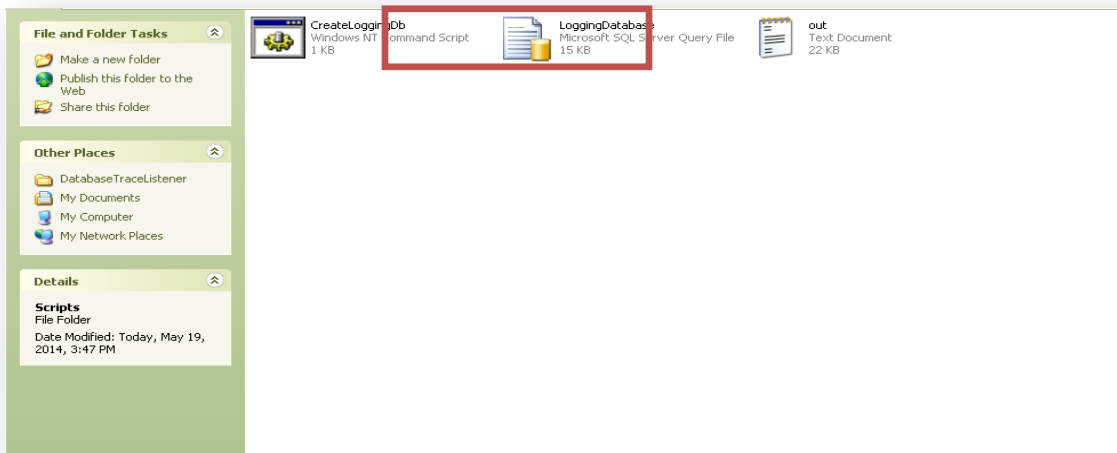


Now it's time to decide which way we have to adopt for logging.
Here we will discuss all the ways.

II. Database Logging

In database logging, we log all the exceptions in Database and for this purpose we execute a predefined script provided in enterprise library source code.

The script can be found at the following location if source code installer is installed in its default location:
C:\DocumentsAndSettings\MyDocuments\EntLib50Src\Blocks\Logging\Src\DatabaseTraceListener\Scripts



Here we find the script with name: **LoggingDatabase**

Basically, the script is nothing but a set of statements to create a database named “Logging” and required stored procedure and table to create in the same database.

After executing the same script, we will find the database “Logging” created in our database set as shown below in the screen shot.

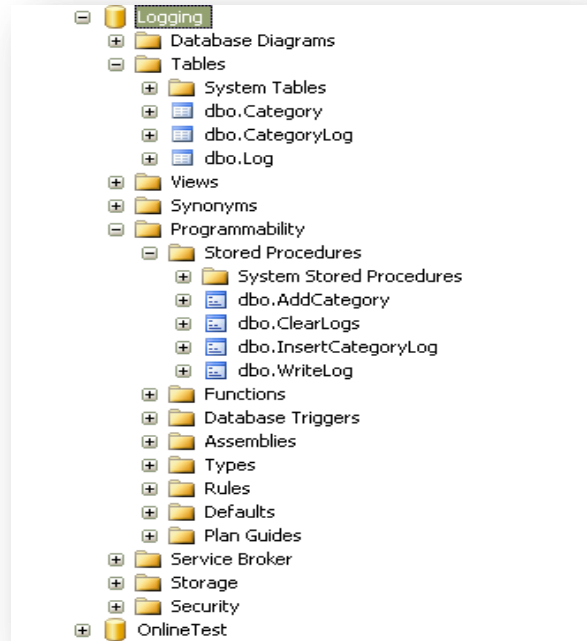
The different tables which are created in the database Logging are as follows:

- Category
- CategoryLog
- Log

The stored procedures created are as follows:

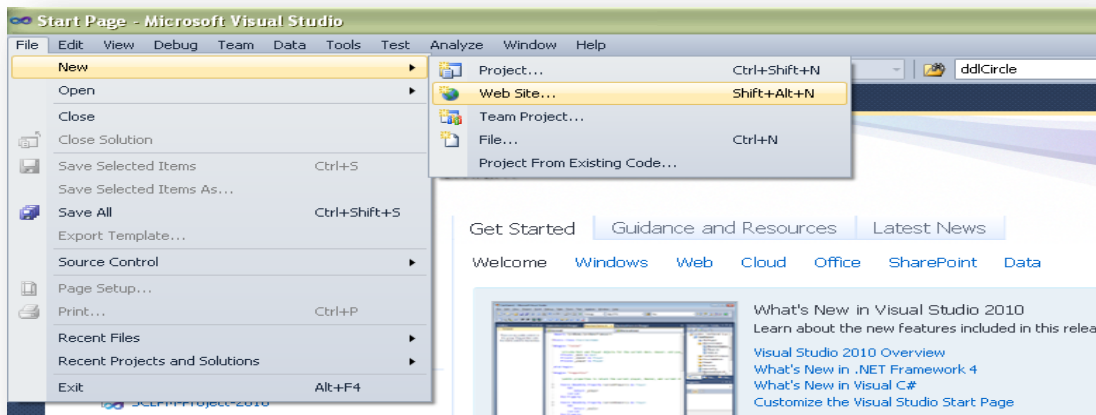
- AddCategory
- ClearLogs
- InsertCategoryLog
- WriteLog

Here the table “Log” will be used to log all the runtime exceptions. Now it’s time to create an application which will throw a runtime exception.

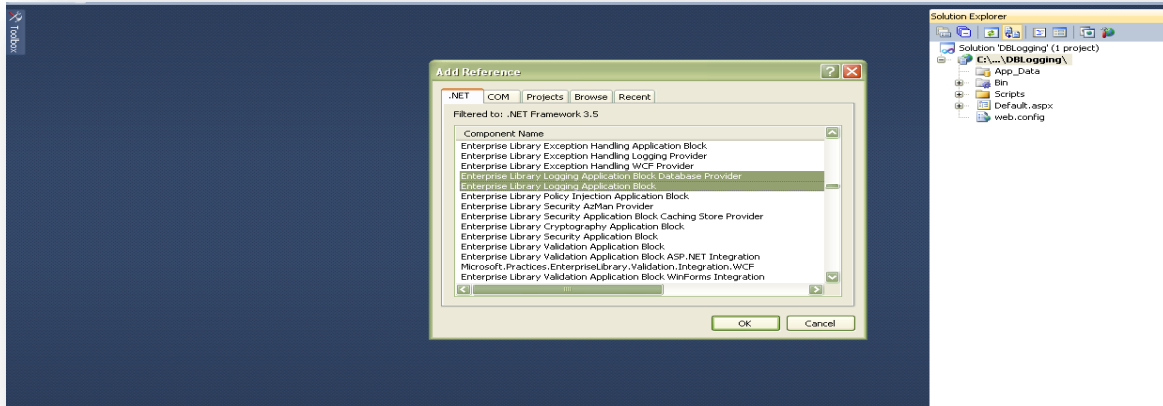


Simply, go to New > Website and Provide the name to the website.

Take a button type on the web page and write a code that will throw an exception. In my code, I have thrown my own custom exception “Anubhav Raised this exception!!”



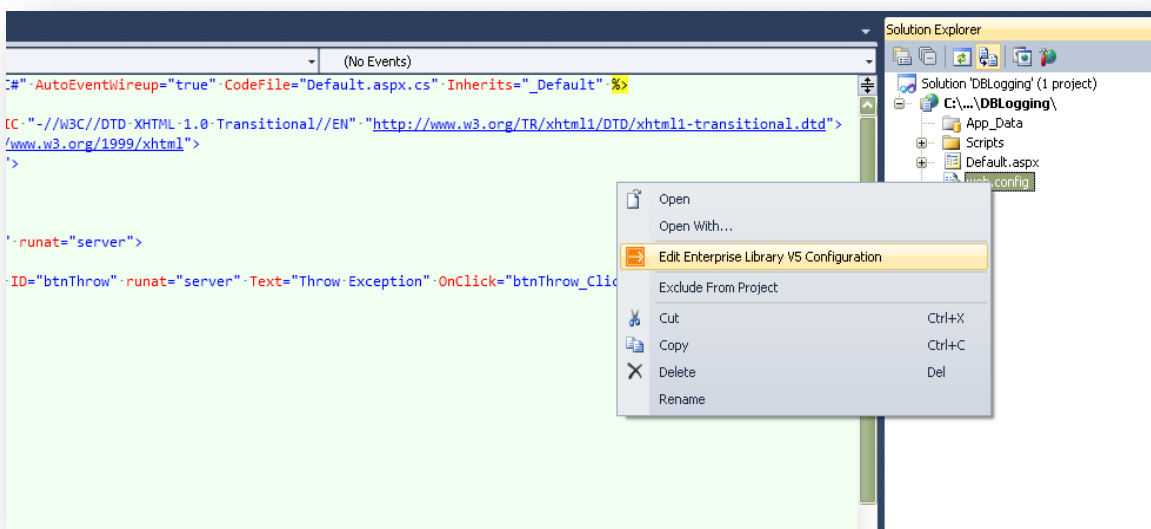
Now add the reference to the Logging Application Block and Logging Application Database Provider so as to avail the same in the web application.



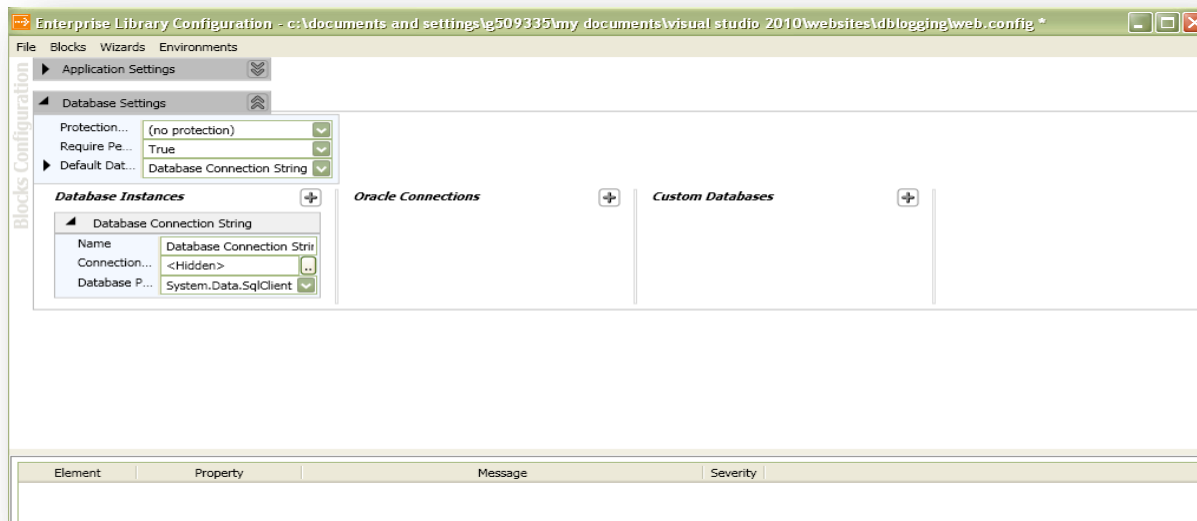
Add the namespace Microsoft.Practices.EnterpriseLibrary.Logging in .cs page.
Now in the catch block, write the code to log the error. The code should be like this

```
try
{
    throw new Exception("Anubhav Raised this exception!!");
}
catch (Exception ex)
{
    LogEntry entry = new LogEntry();
    entry.Message = ex.Message;
    Logger.Write(entry);
}
```

Now use Enterprise Library Configuration Editor to apply the logging setting. This tool is very helpful to provide the settings in web.config. Simply, Right click on Web.config and click on Edit Enterprise Library V5 Configuration.



This will open a window for configuration containing all the setting like Application Setting, Database Settings etc. Moreover, we can add different application blocks for settings like Logging Application Block, Data Access Application Block etc.



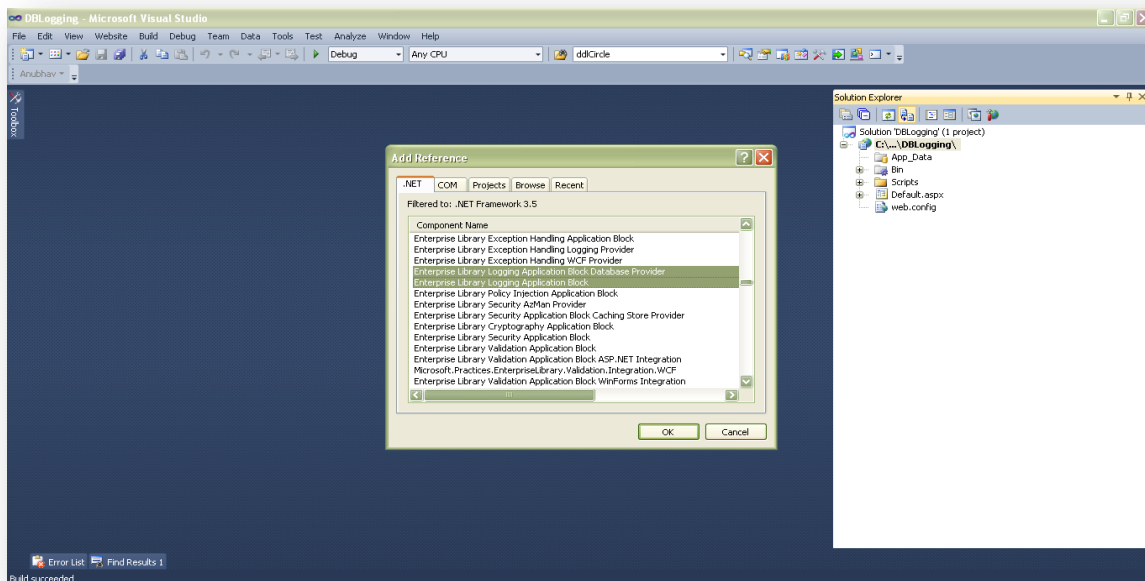
Here we are concerned about Logging Application Block of enterprise library, so we will add Logging setting from the Blocks navigation.

Block >> Add Logging Settings

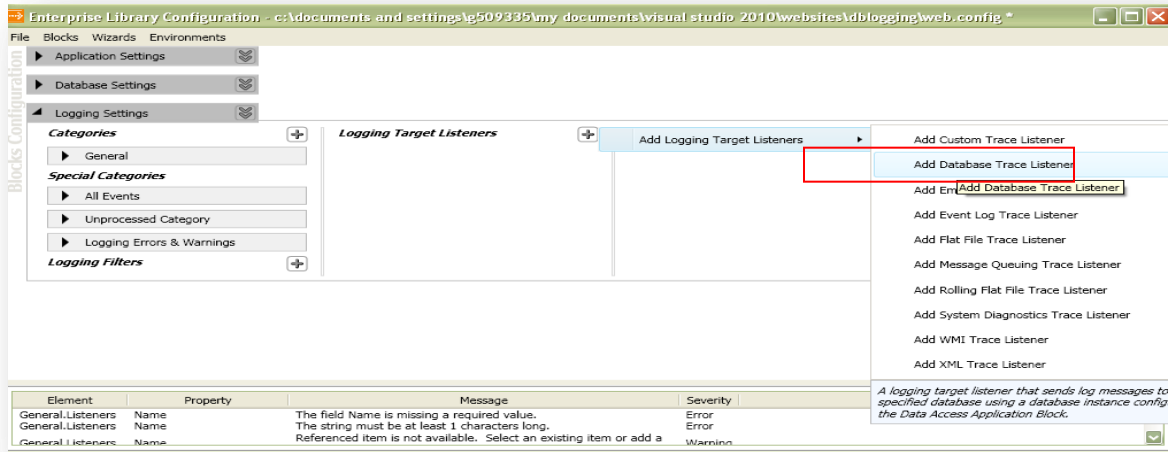


If we want to implement Database Logging we need to add the references of the followings:

- Enterprise Logging Application Block Database Provider
- Enterprise Logging Application Block

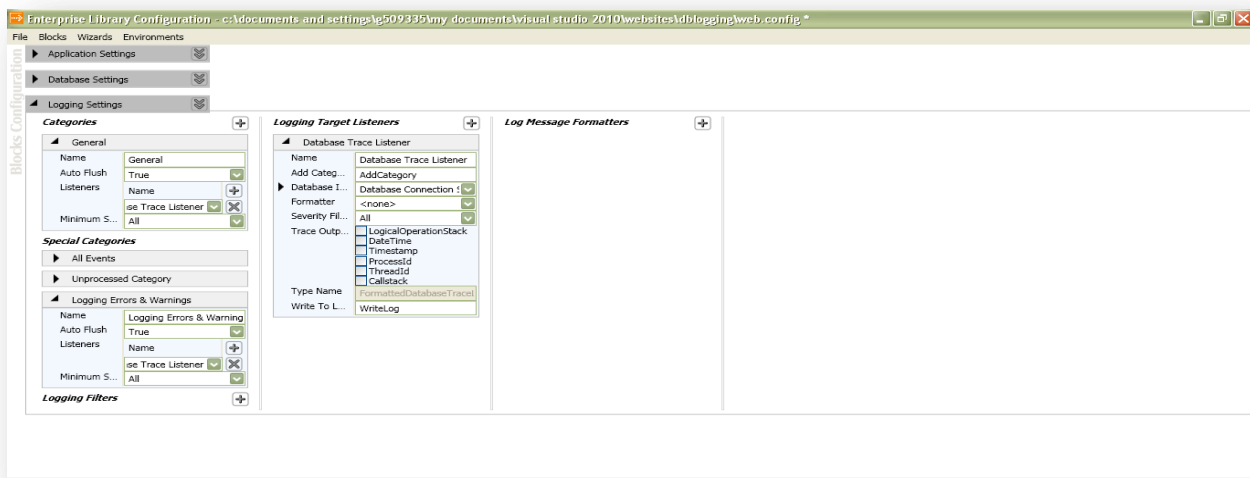


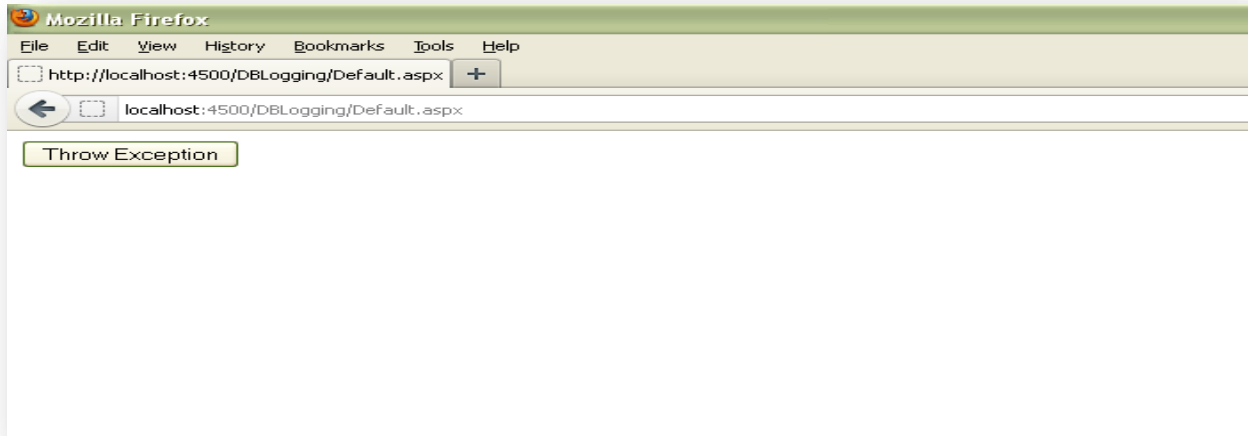
Moreover, we need to modify "web.config" accordingly. For modifying web.config, we use the Enterprise Library configuration editor as shown below in the screen shot: Adding the Logging Trace listener is the most important step for Error Logging.



Following are the properties which must be set through this editor:

- Name: Name for the trace listener.
- Add Category Procedure: The name of the stored procedure in the database that creates a new category in tables.
- Database Instance:
- Name: Name of the connection string.
- Connection String: Entire connection string.
- Database Provider: Database provider name like System.Data.SqlClient
- Formatter: Formatter name like Text Formatter, Binary Formatter etc
- Severity Filter: The minimum severity of messages that will be sent to the logging target.
- Trace Output Options: Use this property to determine which option or elements to be included in the trace output.
- Write to Log Procedure: The name of the stored procedure that writes the log messages into the tables.





Now it's time to verify. Let's click on "Throw Exception" and see the result.

We can clearly look in the table "Log" belonging to the database "Logging" having entries of error messages with different information:

- Severity
- Timestamp
- Process ID
- Thread Name
- App Domain Name
- Message etc

A screenshot of Microsoft SQL Server Management Studio (SSMS) showing a table named 'Log' in the 'dbo' schema of the 'SCL-DBTESTPDR...ging' database. The table contains 11 rows of log entries. The columns are: LogID, EventID, Priority, Severity, Title, Timestamp, MachineName, AppDomainName, ProcessID, ProcessName, ThreadName, Win32ThreadId, and Message. The first 10 rows show information messages with a priority of -1, and the 11th row shows null values for all fields.

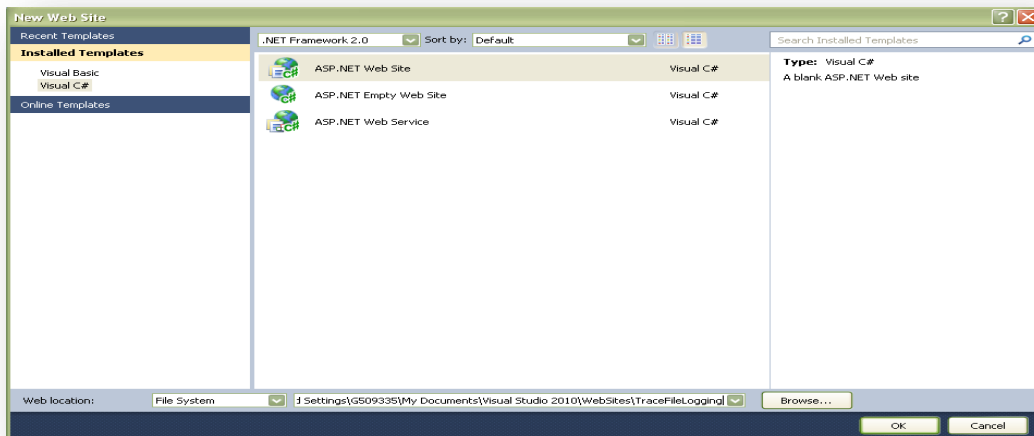
| LogID | EventID | Priority | Severity | Title | Timestamp | MachineName | AppDomainName | ProcessID | ProcessName | ThreadName | Win32ThreadId | Message |
|-------|---------|----------|-------------|-------|-------------------|---------------|-------------------|-----------|---------------------|------------|---------------|---------------------|
| 1 | 0 | -1 | Information | | 2014-05-19 11:... | NDAPC00000109 | 7e9f106-4-1304... | 30336 | C:\Program Files... | NULL | 32484 | Anubhav Raised this |
| 2 | 0 | -1 | Information | | 2014-05-19 11:... | NDAPC00000109 | 7e9f106-4-1304... | 30336 | C:\Program Files... | NULL | 32484 | Anubhav Raised this |
| 3 | 0 | -1 | Information | | 2014-05-19 11:... | NDAPC00000109 | 7e9f106-4-1304... | 30336 | C:\Program Files... | NULL | 32484 | Anubhav Raised this |
| 4 | 0 | -1 | Information | | 2014-05-19 11:... | NDAPC00000109 | 7e9f106-4-1304... | 30336 | C:\Program Files... | NULL | 32484 | Anubhav Raised this |
| 5 | 0 | -1 | Information | | 2014-05-19 11:... | NDAPC00000109 | 7e9f106-4-1304... | 30336 | C:\Program Files... | NULL | 32484 | Anubhav Raised this |
| 6 | 0 | -1 | Information | | 2014-05-19 11:... | NDAPC00000109 | 7e9f106-4-1304... | 30336 | C:\Program Files... | NULL | 32484 | Anubhav Raised this |
| 7 | 0 | -1 | Information | | 2014-05-19 11:... | NDAPC00000109 | 7e9f106-4-1304... | 30336 | C:\Program Files... | NULL | 32484 | Anubhav Raised this |
| 8 | 0 | -1 | Information | | 2014-05-19 11:... | NDAPC00000109 | 7e9f106-4-1304... | 30336 | C:\Program Files... | NULL | 32484 | Anubhav Raised this |
| 9 | 0 | -1 | Information | | 2014-05-19 11:... | NDAPC00000109 | 7e9f106-4-1304... | 30336 | C:\Program Files... | NULL | 32484 | Anubhav Raised this |
| 10 | 0 | -1 | Information | | 2014-05-19 11:... | NDAPC00000109 | 7e9f106-4-1304... | 30336 | C:\Program Files... | NULL | 32484 | Anubhav Raised this |
| 11 | 0 | -1 | Information | | 2014-05-19 11:... | NDAPC00000109 | 7e9f106-4-1304... | 30336 | C:\Program Files... | NULL | 32484 | Anubhav Raised this |
| * | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

III. Flat File Logging

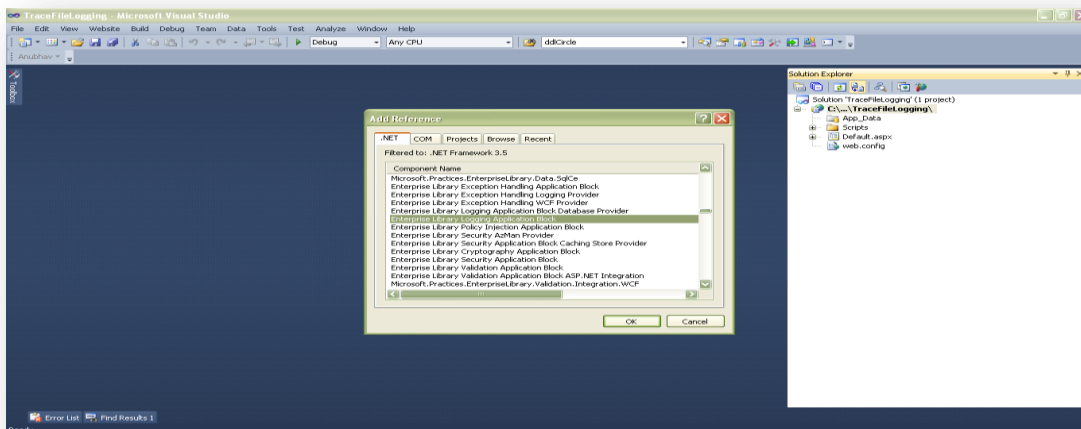
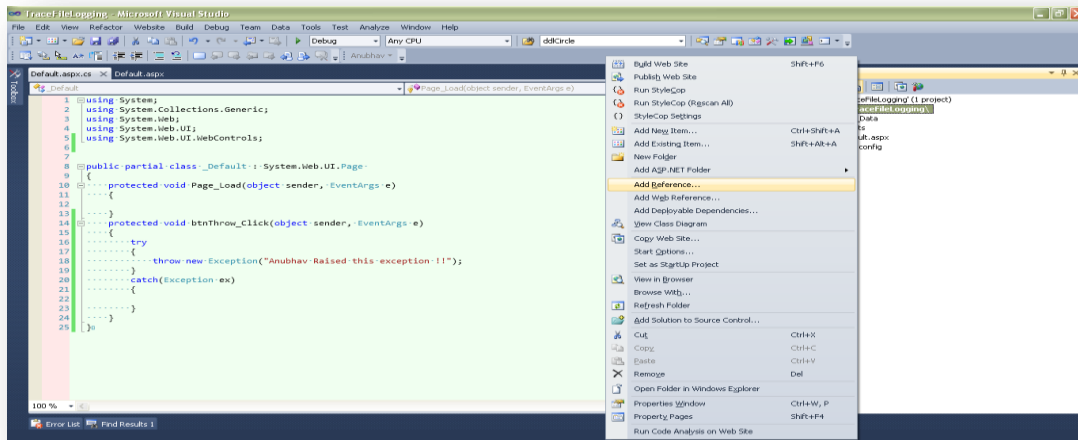
Let's create a new website. I named it as "TraceFileLogging".

All the steps are again same which are as follows:

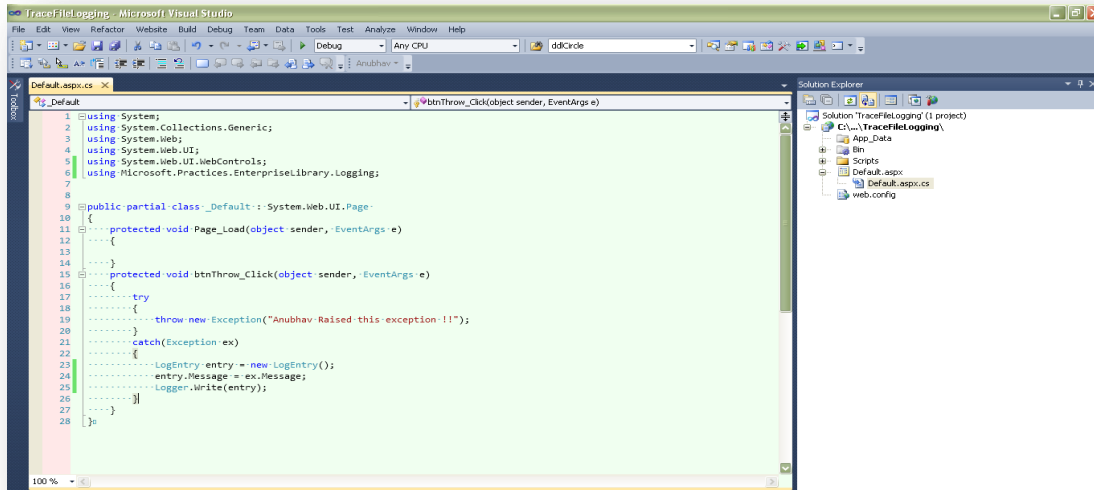
Step.1 Add new web application from VS2010



Step.2 Add reference to Enterprise Library Logging Application Block

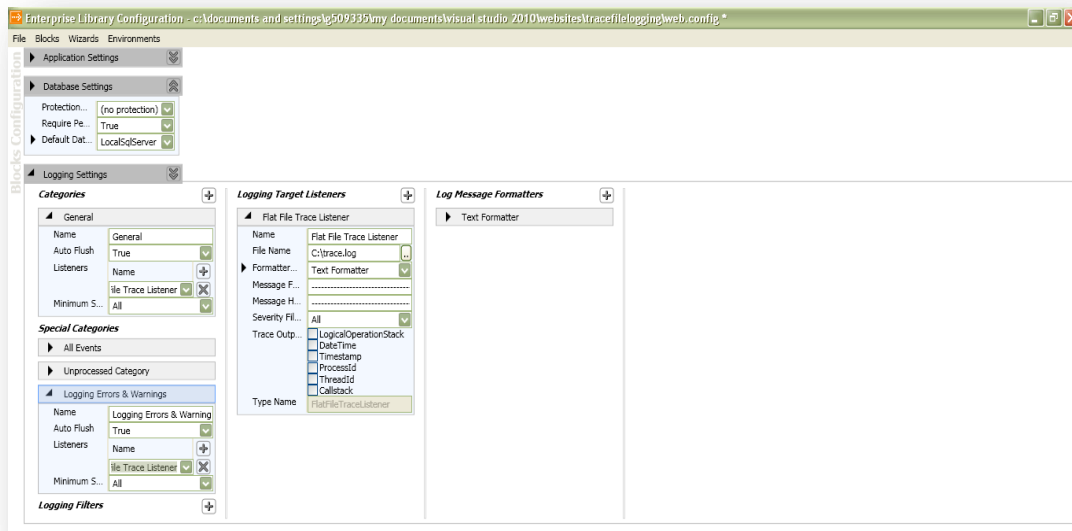


Step.3 Write code for error logging in catch block of .cs page.



```
try
{
    throw new Exception("Anubhav Raised this exception!!");
}
catch (Exception ex)
{
    LogEntry entry = new LogEntry();
    entry.Message = ex.Message;
    Logger.Write(entry);
}
```

Step.4 Modify web.config accordingly using Enterprise Library Configuration Editor

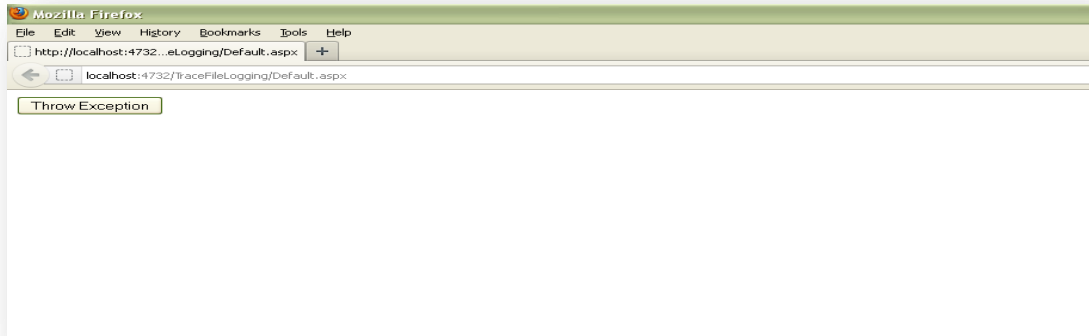


Following are the properties which must be set through this editor:

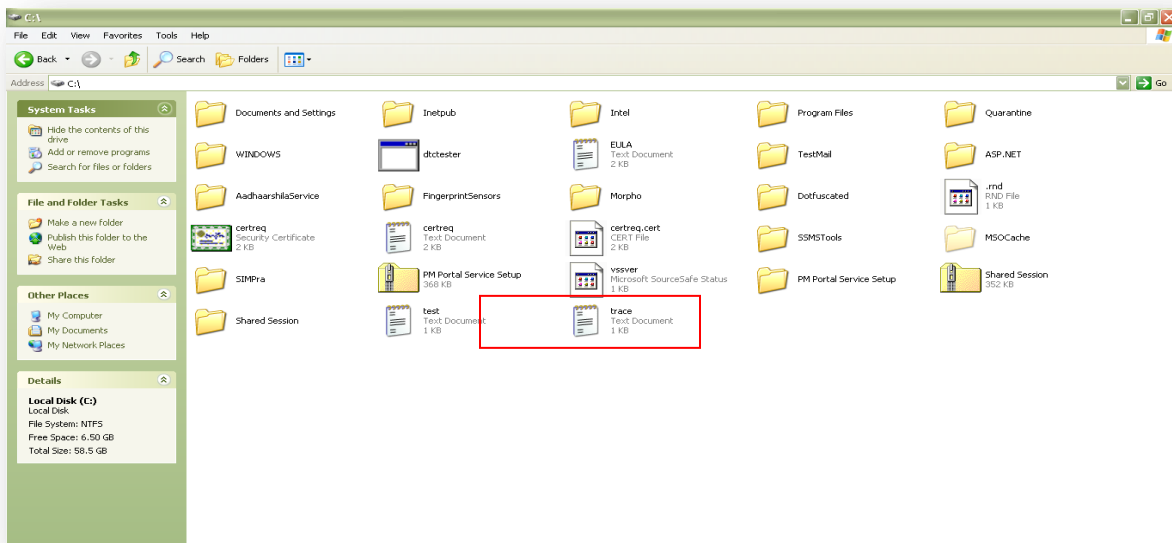
- Name: Name for the trace listener.
- File Name: The path and file name to log the messages.
- Formatter Name:
- Name: Name of the Formatter.
- Template: This is the template used to log the messages.

- Message Footer: The text of the footer to add to the log messages.
- Message Header: The text of the header to add to the log messages.
- Severity Filter: The minimum severity of messages that will be sent to the logging target.
- Trace Output Options: Use this property to determine which option or elements to be included in the trace output.

Step.5 Test your application.



Let's click on "Throw Exception" and see the error log file is created in the respective directory we mentioned for log messages. In my application I provide the file name "C:\trace.txt" and now this log file is available at the location



Let open it to see how the messages are logged. Yes, its done ☺

I am looking that following fields are mentioned in this log file:

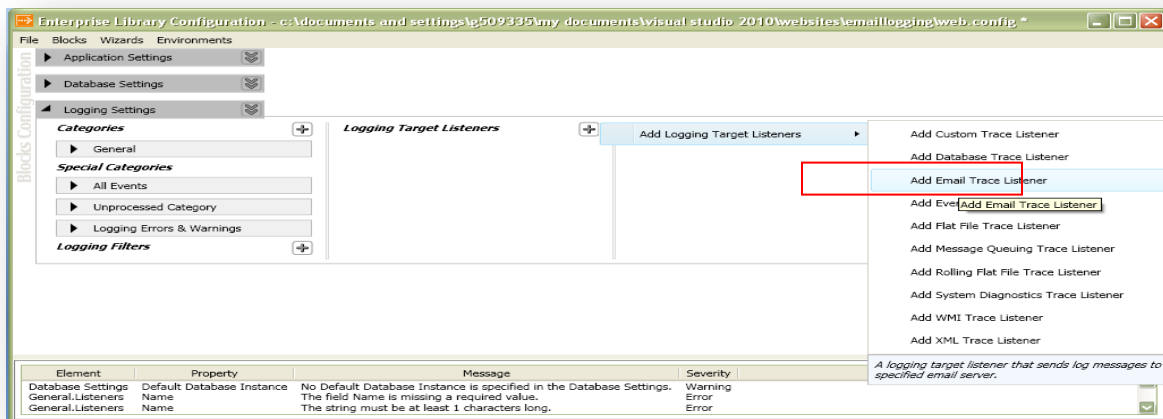
- Timestamp
- Message
- Category
- Priority
- Machine
- Thread Name etc.



IV. Email Logging

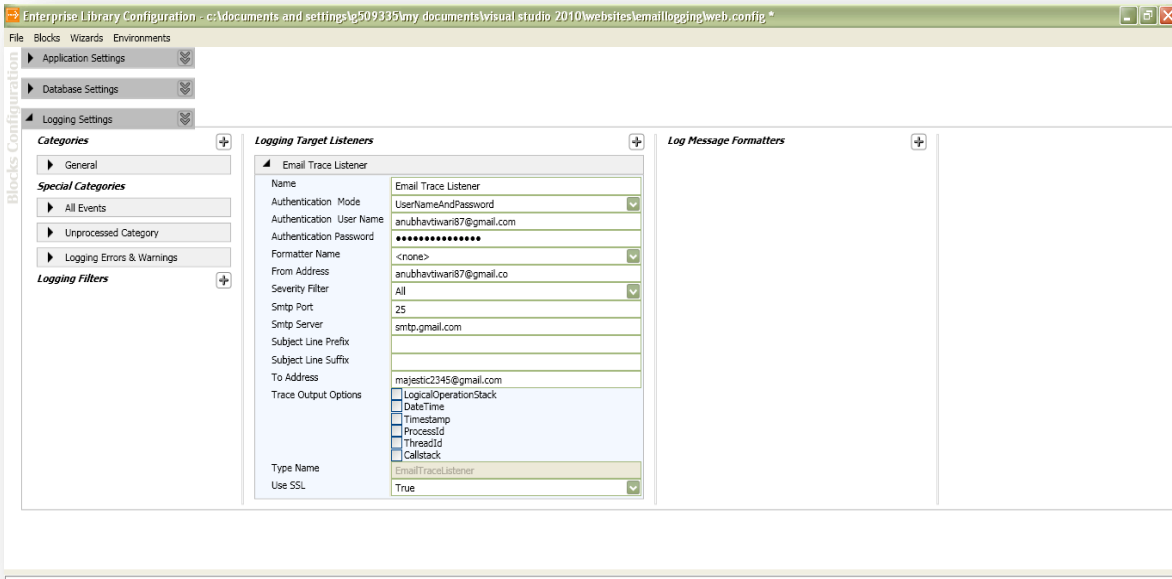
For email logging, all the steps are same as above except the web.config settings.

So I am including the config section editing only. In the configuration editor, we add the Email Trace Listener as show in the screen shot below:

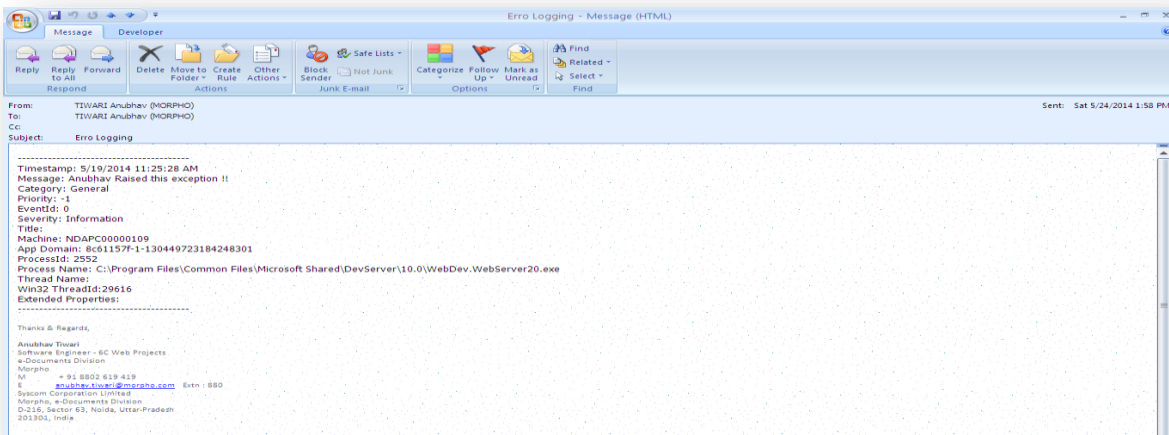


Following are the properties which must be set through this editor:

- Name: Name for the trace listener.
- Authentication Mode: The Mode of server authentication like, username and password, windows authentication etc.
- Authentication Username: The username for authentication.
- Authentication Password: The password for authentication.
- From Address: The from address to use for the email messages created for each log entry.
- Severity Filter: The minimum severity of messages that will be sent to the logging target.
- Smtip Port: The port used to sent the email messages.
- Smtip Server: The name of IP Address of the server that will send the email messages.
- To address: The “To” address to use for the email messages created for each log entry
- Trace output options: Use this property to determine which option or elements to be included in the trace output.
- Use SSL: The Boolean value that specifies if the listener will authenticate and send messages using secure socket layer encrypted communication.

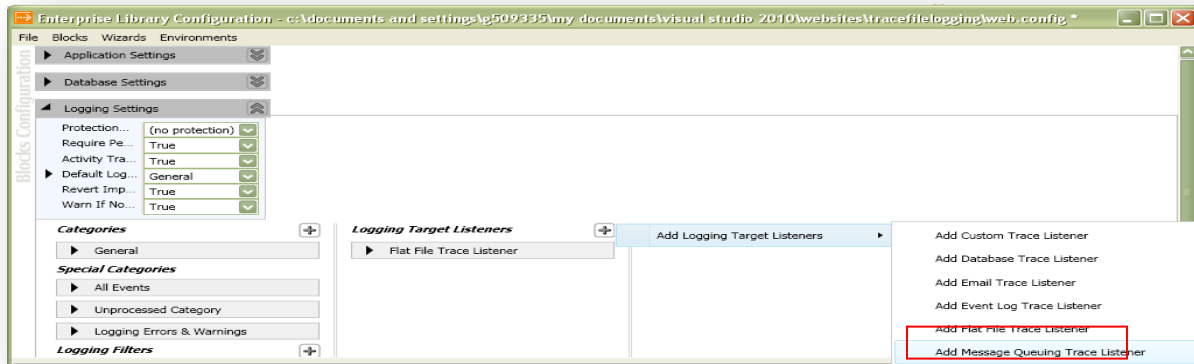


Now let's test the application. After throwing the exception, we receive a mail regarding the error log.



V. Message Queue Logging

For Message Queue logging, all the steps are same as above except the web.config settings. For this we need to add Message Queuing Trace Listener keeping in mind that a message queue is already created in windows as given in the screen shot below:



Rests of the steps are very same as the other logging types mentioned above.

VI. Advantage of using Enterprise Logging

For the database application block, the main advantage is that it makes it easier to produce database-agnostic code. The developer interacts mainly with generic Database and Db Command objects, rather than e.g. SqlConnection, SqlCommand, etc. Thus, switching to a different database (i.e. Oracle) becomes more feasible.

Enterprise Library also gently reduces the risk of SQL injection attacks.

Exception and Logging together make it very easy to log exceptions.

It's highly configurable, and if we use it well we'll be able to log comprehensively and with great flexibility.

VII. CONCLUSION

Thus we can conclude that Enterprise Library is really very helpful to the developers for handling and identifying the exceptions/errors which are very difficult on production environment. The use of it is very simple as described in the paper.

ACKNOWLEDGMENT

This work was supported by Syscom Corporation Ltd.

REFERENCES

- [1] <http://www.codeproject.com/Articles/239878/Distributed-Logging-thru-Enterprise-Library-EntL>
- [2] <http://www.codeproject.com/Articles/9545/Get-Logging-with-the-Enterprise-Library>.

AUTHORS

First Author – Anubhav Tiwari, M.C.A, Software Engineer, Anubhav.tiwari@morpho.com.