**Extending Log4Net for Logging Custom data objects**

Log4Net library is used for purposes of application debugging and auditing. This library has its own defined log format that can be logged to the database or to the text files. But Most of the time there could be requirement to log custom messages that provides logging capability to store information about the actions users are performing on an application.

This article would help anyone try to implement log4net on their application for the purpose of user activity tracking, stored in an MSSQL server database.

**Step 1:** First we would need to implement a class that will hold activity properties to be logged. Let call this class as ActionLoggerInfo. This class will need to override ToString() method to serialize the object to Json.

class ActionLoggerInfo

{

public string Module { get; set; }

public DateTime DateTime { get; set; }

public string Status { get; set; }

public string Message { get; set; }

public int UserId { get; set; }

public string IPAddress { get; set; }

public override string ToString()

{

return JsonConvert.SerializeObject(this);

}

}

**Step 2:** Here we would need to create custom Conversion Pattern that would be used by each respective appenders. E.g. RollingFileAppender, ADONetAppender

public class ActionLayoutPattern : PatternLayout

{

public ActionLayoutPattern()

{

AddConverter(new ConverterInfo

{

Name = "actionInfo",

Type = typeof(ActionConverter)

}

);

}

}

**Step 3:** Next to implement a converter is responsible for handling conversion requests of a LoggingEvent when log4net

public class ActionConverter : PatternConverter

{

protected override void Convert(System.IO.TextWriter writer, object state)

{

if (state == null)

{

writer.Write(SystemInfo.NullText);

return;

}

var loggingEvent = state as LoggingEvent;

var actionInfo = loggingEvent.MessageObject as ActionLoggerInfo;

if (actionInfo == null)

{

writer.Write(SystemInfo.NullText);

}

else

{

switch (this.Option.ToLower())

{

case "module":

writer.Write(actionInfo.Module);

break;…

**Step 4:** Next step would be to configure log4net. Create table for storing the logs. Then you would need to configure appneder’s to use custom action logger. While specifying the parameters mentioned. For example,

<layout type="WPFLog4Net.ActionLayoutPattern">

<conversionPattern value="%actionInfo{module}" />

</layout>

That is it your, application should now start logging your custom logs to your data store. Please refer attached sample application for more details.

Reference:

<https://logging.apache.org/log4net/release/sdk/log4net.Layout.PatternLayout.html>

<http://www.hanselman.com/blog/CreatingYourOwnCustomPatternLayoutPatternParserAndPatternConvertorWithLog4net.aspx>

Log4net,log4net custom log,log,error log