**NLog Presentation Notes**

* **How to setup and configure it in a project**

*<configuration>*

*<configSections>*

*<section name=”nlog” type=”NLog.Config.ConfigSectionHandler, NLog” />*

*</configSections>*

*<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">*

*</nlog>*

*</configuration>*

* **What is a target?**

Where you are logging to

*<targets>*

*<target name=”f1” xsi:type=”File” fileName=”generalDebugStuff.txt”/>*

*<target name=”f2” xsi:type=”File” fileName=”errorFileThing.txt”/>*

*</targets>*

* **How do you configure logging to specific targets?**

There are multiple ways:

Console

*<target name=”consolTarg” xsi:type=”Console”/>*

Email

*<target name=”emalTarg” xsi:type=”Mail”*

*to=”blah@blah.com”*

*from=”otherblah@blah.com”*

*cc=”yourblah@blah.com”*

*subject=”This is a Blah message at time ${longdate}”*

*body=”Something happened, here it is: ${longdate} ${callsite} ${level} ${message}“ />*

Rolling File

*<target name=”file” xsi:type=”File” layout=”${longdate} ${logger} ${message}” fileName=”generalDebugStuff.txt”* ***archiveFileName=”generalDebugStuff.{##}.txt” archiveAboveSize=”10240” archiveNumbering=”Rolling”*** *keepFileOpen=”false” />*

* **What is a rule?**

This determines which targets to use when logging, based on a log level (Error, Info)

*<rules>*

*<logger name=”Name.Space.\*” minlevel=”Debug” writeTo=”f1”/>*

*<logger name=”Name.Space.Class1” levels=”Error, Fatal” writeTo=”f2”/>*

*</rules>*

* **What are Layout Renderers?**

These determine how each line in a log is formatted and what information is displayed, Examples:

Date, class name

On your target element:

*<target name=”f1” xsi:type=”File” fileName=”generalDebugStuff.txt” layout=”${longdate} ${callsite} ${level} ${message}” />*

* **How do you configure it to log with UTC?**

Using date inside your Layout Renderer

*${date:universalTime=true}*

* **How Often Should files rollover?**

It depends on how much you are logging

If you are worried about file sizes, you would roll files over based on data amount,

If you are worried about number of logs, you could instead roll a file over based on time

* **Async target: when to use it**

When you are logging a large amount of data, but you don’t want your code to wait around for logging things, when you logging on a UI thread.

**Async target: How to use it**

**Asynchronous file logging**

*<target name=”f1” xsi:type=”AsyncWrapper” queueLimit=”5000” overflowAction=”Discard”>*

*<target xsi:type=”File” fileName=”generalDebugStuff.txt” />*

*</target>*

**Or**

*<targets async=”true”>*

*<target xsi:type=”File” fileName=”generalDebugStuff.txt” />*

*</targets>*

**How to configure a custom target?**

*[Target("TypeDebugCustomTargThingy")]*

*public class TypeDebugCustomTarget : TargetWithLayout*

*{*

*public TypeDebugCustomTarget()*

*{*

*File = "debugFile.txt";*

*}*

*[RequiredParameter]*

*public string File { get; set; }*

*protected override void Write(LogEventInfo logEvent)*

*{*

*var logMessage = this.Layout.Render(logEvent);*

*WriteMessageToFile(this.File, logMessage);*

*}*

*private void WriteMessageToFile(string file, string message)*

*{*

*using (var fileStream = new FileStream(file, FileMode.OpenOrCreate))*

*{*

*using (var streamWriter = new StreamWriter(fileStream))*

*{*

*streamWriter.WriteLine(message);*

*}*

*}*

*}*

*}*

**In Config File:**

*<nlog>*

*<extensions>*

*<add assembly="TypeCalculator"/>*

*</extensions>*

*<targets>*

*<target name=”customDebugTarg” xsi:type=”TypeDebugCustomTargThingy” file=”logs/debugWithCustomThing.txt”/>*

**How to use it in code:**

* **LogManager.GetLogger(name)**

Name is what you put in the configuration file, which is usually the class name

For example:

I configure a logger

<logger name=”Blah.Namespace.Blah” writeTo=”blahTarget” />

Then I pass the name “Blah.Namespace.Blah” to the GetLogger method or you could use GetCurrentClassLogger

* **NLog.Interface.ILogger through the Container**

You can get this from NLog.Interface

This is useful when you don’t want to tie your project to NLog in a bunch of different ways, instead there would only be one interface to go through, this will enable you to switch what logging API you use easier.

* **Logger names**

The name is usually the class name, this is used to find a <logger> within your configuration file. This will be put into a log when you use ${logger}

* **Logging Exceptions**

Try {

Code that blows up

}

Catch(Exception e) {

var logger = LogManager.GetCurrentClassLogger()

logger.Error(“Oh no, an exception happened: ”, e)

}

**Common Usages:**

* **How do I include the log level in the layout?**

Just add it to the target:

*<target name=”f1” xsi:type=”File” fileName=”generalDebugStuff.txt”* ***layout=”${level}”>***

* **How do I rollover the log file every day and beyond a certain size?**

Just add it to the target:

*<target name=”f1” xsi:Type=”File” fileName=”generalDebugStuff.${date}.txt”* ***archiveFileName=”generalDebugStuff.{##}.txt” archiveAboveSize=”10240” archiveNumbering=”Rolling” keepFileOpen=”false”*** */>*

* **How do I format exceptions in the layout?**

Like this:

*${exception:format=tostring}*

**What to include when logging?**

Things like Date, Time, Class, Exception Message, Stack Trace, Object Identifiers

**What shouldn’t be logged?**

Things that not are unique, Namespaces, Large sets of data, Things that won’t help when debugging, such as “Error Happened”, Repetitive Tasks such as putting things into SQL

**Where should log files go?**

They should be in a place easy to access or be referenced by the Devs such as on a network share

**Integration with LogStash**

**How do you send messages to LogStash via TCP?**

You just add it to your target:

*<target name=”f3”* ***xsi:type=”Network”*** *onOverflow=”Discard” newLine=”true” layout=”Whatever Layout you Want: ${longdate}”* ***address=”tcp://yourLogStashHost:yourLogStashPort”/>***