Apache Log4Net (Ver 2.0)

Background

The Apache Log4net Analysis App presents a predefined set of dashboards and gadgets visualizing Log4net logs. The Log4net analysis pack addresses the need to manage and debug Apache Log4net applications and infrastructure during development, testing, and production. This App helps measure, troubleshoot, and optimize Java based applications with visualization and investigation dashboards.

Steps

1. Add Log Data In XpoLog, When adding a log to XpoLog you can now select the Log Type (logtype) for Apache log4net with the following logtype:
   a. *log4net*
2. Once all required information is set click next and edit the log pattern, this step is crucial to the accuracy and deployment of the App. Use the following conversion table to build the XpoLog pattern out of the log4net log format.

Example

In the Apache Log4Net configuration file, can be either properties files, XML file, or in some case the log format was created pro-grammatically for which you can manually create the pattern for the data.

```
<appender name="A1" type="log4net.Appender.ConsoleAppender">
    <!-- A1 uses PatternLayout -->
    <layout type="log4net.Layout.PatternLayout">
            <conversionPattern value="%timestamp [%thread] %-5level %logger %ndc - %message%newline" />
    </layout>
</appender>
```

The following sequence is the log structure definition for the log4net log %timestamp [%thread] %-5level %logger %ndc - %message%newline

In XpoLog such pattern will be translated into:

for more information see below:

{date:Date,locale=en,yyyy-MM-dd HH:mm:ss,SSS} [{text:Thread,ftype=thread}] [{priority:Priority,ftype=severity;,DEBUG;INFO;WARNING;ERROR;FATAL}] {string:Logger,ftype=logger} {string:NDC,ftype=ndc} - {string:Message,ftype=message}

Apache Log4Net Conversion Table

logtyep should be set to: *log4net*

Name and Appears with	Description	XpoLog Pattern	ftype
a	Equivalent to **appdomain**		
appdomain	Used to output the friendly name of the AppDomain where the logging event was generated.		
aspnet-cache	Used to output all cache items in the case of **%aspnet-cache** or just one named item if used as **%aspnet-cache{key}** This pattern is not available for Compact Framework or Client Profile assemblies.		
aspnet-context	Used to output all context items in the case of **%aspnet-context** or just one named item if used as **%aspnet-context{key}** This pattern is not available for Compact Framework or Client Profile assemblies.		
aspnet-request	Used to output all request parameters in the case of **%aspnet-request** or just one named param if used as **%aspnet-request{key}** This pattern is not available for Compact Framework or Client Profile assemblies.		

aspnet-session	Used to output all session items in the case of **%aspnet-session** or just one named item if used as **%aspnet-session{key}** This pattern is not available for Compact Framework or Client Profile assemblies.		
c	Equivalent to **logger**		
C	Equivalent to **type**		
class	Equivalent to **type**		
d	Equivalent to **date**		
date	Used to output the date of the logging event in the local time zone. To output the date in universal time use the %utcdate pattern. The date conversion specifier may be followed by a *date format specifier* enclosed between braces. For example, **%date{HH:mm:ss,fff}** or **%date{dd MMM yyyy HH:mm:ss,fff}**. If no date format specifier is given then ISO8601 format is assumed (Iso8601DateFormatter). The date format specifier admits the same syntax as the time pattern string of the **ToString**. For better results it is recommended to use the log4net date formatters. These can be specified using one of the strings "ABSOLUTE", "DATE" and "ISO8601" for specifying AbsoluteTimeDateFormatter, DateTimeDateFormatter and respectively **Iso8601DateFormatter**. For example, **%date{ISO8601}** or **%date{ABSOLUTE}**. These dedicated date formatters perform significantly better than **ToString**.		
exception	Used to output the exception passed in with the log message. If an exception object is stored in the logging event it will be rendered into the pattern output with a trailing newline. If there is no exception then nothing will be output and no trailing newline will be appended. It is typical to put a newline before the exception and to have the exception as the last data in the pattern.		
F	Equivalent to **file**		
file	Used to output the file name where the logging request was issued. **WARNING** Generating caller location information is extremely slow. Its use should be avoided unless execution speed is not an issue. See the note below on the availability of caller location information.		
identity	Used to output the user name for the currently active user (Principal.Identity.Name). **WARNING** Generating caller information is extremely slow. Its use should be avoided unless execution speed is not an issue.		
l	Equivalent to **location**		
L	Equivalent to **line**		
location	Used to output location information of the caller which generated the logging event. The location information depends on the CLI implementation but usually consists of the fully qualified name of the calling method followed by the callers source the file name and line number between parentheses. The location information can be very useful. However, its generation is **extremely** slow. Its use should be avoided unless execution speed is not an issue. See the note below on the availability of caller location information.		
level	Used to output the level of the logging event.		
line	Used to output the line number from where the logging request was issued. **WARNING** Generating caller location information is extremely slow. Its use should be avoided unless execution speed is not an issue. See the note below on the availability of caller location information.		
logger	Used to output the logger of the logging event. The logger conversion specifier can be optionally followed by *precision specifier*, that is a decimal constant in brackets. If a precision specifier is given, then only the corresponding number of right most components of the logger name will be printed. By default the logger name is printed in full. For example, for the logger name "a.b.c" the pattern **%logger{2}** will output "b.c".		
m	Equivalent to **message**		
M	Equivalent to **method**		
message	Used to output the application supplied message associated with the logging event.		

mdc	The MDC (old name for the ThreadContext.Properties) is now part of the combined event properties. This pattern is supported for compatibility but is equivalent to **property**.		
method	Used to output the method name where the logging request was issued.		
	WARNING Generating caller location information is extremely slow. Its use should be avoided unless execution speed is not an issue.		
	See the note below on the availability of caller location information.		
n	Equivalent to **newline**		
newline	Outputs the platform dependent line separator character or characters.		
	This conversion pattern offers the same performance as using non-portable line separator strings such as "\n", or "\r\n". Thus, it is the preferred way of specifying a line separator.		
ndc	Used to output the NDC (nested diagnostic context) associated with the thread that generated the logging event.		
p	Equivalent to **level**		
P	Equivalent to **property**		
properties	Equivalent to **property**		
property	Used to output the an event specific property. The key to lookup must be specified within braces and directly following the pattern specifier, e.g.**%property{user}** would include the value from the property that is keyed by the string 'user'. Each property value that is to be included in the log must be specified separately. Properties are added to events by loggers or appenders. By default the log4net:HostName property is set to the name of machine on which the event was originally logged.		
	If no key is specified, e.g. **%property** then all the keys and their values are printed in a comma separated list.		
	The properties of an event are combined from a number of different contexts. These are listed below in the order in which they are searched.		
	the event properties		
	The event has Properties that can be set. These properties are specific to this event only.		
	the thread properties		
	The Properties that are set on the current thread. These properties are shared by all events logged on this thread.		
	the global properties		
	The Properties that are set globally. These properties are shared by all the threads in the AppDomain.		
r	Equivalent to **timestamp**		
stacktrace	Used to output the stack trace of the logging event The stack trace level specifier may be enclosed between braces. For example, **%stacktrace{level}**. If no stack trace level specifier is given then 1 is assumed		
	Output uses the format: type3.MethodCall3 > type2.MethodCall2 > type1.MethodCall1		
	This pattern is not available for Compact Framework assemblies.		
stacktracedetail	Used to output the stack trace of the logging event The stack trace level specifier may be enclosed between braces. For example, **%stacktracedetail{level}**. If no stack trace level specifier is given then 1 is assumed		
	Output uses the format: type3.MethodCall3(type param,...) > type2.MethodCall2(type param,...) > type1.MethodCall1(type param,...)		
	This pattern is not available for Compact Framework assemblies.		
t	Equivalent to **thread**		
timestamp	Used to output the number of milliseconds elapsed since the start of the application until the creation of the logging event.		
thread	Used to output the name of the thread that generated the logging event. Uses the thread number if no name is available.		
type	Used to output the fully qualified type name of the caller issuing the logging request. This conversion specifier can be optionally followed by *precision specifier*, that is a decimal constant in brackets.		
	If a precision specifier is given, then only the corresponding number of right most components of the class name will be printed. By default the class name is output in fully qualified form.		
	For example, for the class name "log4net.Layout.PatternLayout", the pattern **%type{1}** will output "PatternLayout".		
	WARNING Generating the caller class information is slow. Thus, its use should be avoided unless execution speed is not an issue.		
	See the note below on the availability of caller location information.		

u	Equivalent to **identity**		
username	Used to output the WindowsIdentity for the currently active user. **WARNING** Generating caller WindowsIdentity information is extremely slow. Its use should be avoided unless execution speed is not an issue.		
utcdate	Used to output the date of the logging event in universal time. The date conversion specifier may be followed by a *date format specifier* enclosed between braces. For example, **%utcdate{HH:mm:ss,fff}** or **%utcdate{dd MMM yyyy HH:mm:ss,fff}**. If no date format specifier is given then ISO8601 format is assumed (**Iso8601DateFormatter**). The date format specifier admits the same syntax as the time pattern string of the **ToString**. For better results it is recommended to use the log4net date formatters. These can be specified using one of the strings "ABSOLUTE", "DATE" and "ISO8601" for specifying **AbsoluteTimeDateFormatter**, **DateTimeDateFormatter** and respectively **Iso8601DateFormatter**. For example, **%utcdate{ISO8601}** or**%utcdate{ABSOLUTE}**. These dedicated date formatters perform significantly better than **ToString**.		
w	Equivalent to **username**		
x	Equivalent to **ndc**		
X	Equivalent to **mdc**		
%	The sequence %% outputs a single percent sign.		