log4net配置:log4net独占日志文件的问题以及 log4net的各种输出配置

由于log4net默认情况下会独占日志文件该文件不能被File.Open
可以通过增加配置:<lockingModel type="log4net.Appender.FileAppender+MinimalLock" />来使用最小锁定模型(minimal locking model)以允许多个进程可以写入同个文件

各种appender介绍说明:

在log4net配置中appender是最重要部分般来说每种appender都表示种日志输出介质如日志文件、EvengLog、数据库、控制台、邮件、ASP.NET页面等

本文对各种内置appender配置提供了举例但却远称不上详尽要想了解每种appender参数和选项介绍说明请参看该appenderSDK文档

以下举例都是.NET 2.0下进行, log4net版本为1.2.10

AdoNetAppender
详情参考 log4net.Appender. AdoNetAppender SDK文档

AdoNetAppender相关配置内容取决于目标数据库provider下面仅提供SQL Server 2000例子

首先建立数据表:

```
CREATE TABLE [dbo].[Log]
(
[Id] [] IDENTITY (1, 1) NOT NULL,
[Date] [datetime] NOT NULL,
[Thread] [varchar] (255) NOT NULL,
```

[Level] [varchar] (50) NOT NULL,
[Logger] [varchar] (255) NOT NULL,
[Message] [varchar] (4000) NOT NULL,
[Exception] [varchar] (2000) NULL
)
然后添加配置:

```
<appender name="AdoNetAppender" type="log4net.Appender.AdoNetAppender">
<bufferSize value="2" />
<connectionType value=".Data.SqlClient.SqlConnection, .Data, Version=2.0.0.0, Culture=Neutral,
PublicKeyToken=b77a5c561934e089" />
<connectionString value="server=(local);database=TestBase;egrated security=false;persist security
info=True;UID=sa;PWD=" />
<commandText value="INSERT INTO Log ([Date],[Thread],[Level],[Logger],[Message],[Exception])
VALUES (@log_date, @thread, @log_level, @logger, @message, @exception)" />
<parameter>
<parameterName value="@log_date" />
<dbType value="DateTime" />
<layout type="log4net.Layout.RawTimeStampLayout" />
</parameter>
<parameter>
<parameterName value="@thread" />
<dbType value="String" />
<size value="255" />
<layout type="log4net.Layout.PatternLayout">
<conversionPattern value="%thread" />
</layout>
</parameter>
<parameter>
<parameterName value="@log_level" />
<dbType value="String" />
<size value="50" />
<layout type="log4net.Layout.PatternLayout">
<conversionPattern value="%level" />
```

```
</layout>
</parameter>
<parameter>
<parameterName value="@logger" />
<dbType value="String" />
<size value="255" />
<layout type="log4net.Layout.PatternLayout">
<conversionPattern value="%logger" />
</layout>
</parameter>
<parameter>
<parameterName value="@message" />
<dbType value="String" />
<size value="4000" />
<layout type="log4net.Layout.PatternLayout">
<conversionPattern value="%message" />
</layout>
</parameter>
<parameter>
<parameterName value="@exception" />
<dbType value="String" />
<size value="2000" />
<layout type="log4net.Layout.ExceptionLayout" />
</parameter>
</appender>
```

bufferSize表示批处理日志事件可以避免每次日志事件都访问数据库；ConnectionType指定了要使用IDbConnection完全限定类型名称；connectionString表示连接串；CommandText是SQL语句或存储过程；最后组parameter节点描述了SQL语句或存储过程需要参数

AspNetTraceAppender

详情参考 log4net.Appender.AspNetTraceAppender SDK 文档

<appender name="AspNetTraceAppender" type="log4net.Appender.AspNetTraceAppender" >
<layout type="log4net.Layout.PatternLayout">
<conversionPattern value="%date [%thread] %-5level %logger [%property{NDC}] - %message%line"
/>
</layout>
</appender>
这段配置可将日志信息输出到页面Trace上下文环境如果日志级别低于WARN会以.Web.TraceContext.Write思路方法输出；如果级别为WARN或WARN以上则会以.Web.TraceContext.Warn思路方法输出下图中日志信息区别颜色可以介绍说明这点效果图如下:

这在进行页面调试时候可是很方便
BufferingForwardingAppender

详情参考 log4net.Appender.BufferingForwardingAppender SDK 文档

<appender name="BufferingForwardingAppender"
type="log4net.Appender.BufferingForwardingAppender" >

<bufferSize value="5"/>
<lossy value="true" />
<evaluator type="log4net.Core.LevelEvaluator">
<threshold value="WARN"/>
</evaluator>
<appender-ref ref="LogFileAppender" />
<appender-ref ref="AspNetTraceAppender" />
</appender>
BufferingForwardingAppender主要作用是将输出到指定类型(这里是LogFileAppender)Appender日志信息进行缓存CachebufferSize属性指定了缓存Cache数量如果value为5那么将在信息量达到6条时候把这些日志批量输出appender-ref属性指定了缓存CacheAppender类型同root节点样这里可以指定多个

ColoredConsoleAppender
详情参考log4net.Appender.ColoredConsoleAppender SDK 文档

ColoredConsoleAppender将日志信息输出到控制台默认情况下日志信息被发送到控制台标准输出流下面这个举例演示了如何高亮显示Error信息

```
<appender name="ColoredConsoleAppender" type="log4net.Appender.ColoredConsoleAppender">
<mapping>
<level value="ERROR" />
<foreColor value="White" />
<backColor value="Red, HighIntensity" />
</mapping>
<layout type="log4net.Layout.PatternLayout">
<conversionPattern value="%date [%thread] %-5level %logger [%property{NDC}] - %message%line"
/>
</layout>
</appender>
```

效果如下:

还可以为区别级别指定区别颜色:

```
<appender name="ColoredConsoleAppender" type="log4net.Appender.ColoredConsoleAppender">
<mapping>
<level value="ERROR" />
<foreColor value="White" />
<backColor value="Red, HighIntensity" />
</mapping>
<mapping>
<level value="DEBUG" />
<backColor value="Green" />
</mapping>
<layout type="log4net.Layout.PatternLayout">
<conversionPattern value="%date [%thread] %-5level %logger [%property{NDC}] - %message%line"
/>
</layout>
</appender>
```

效果如下:

ConsoleAppender
详情参考 log4net.Appender.ConsoleAppender SDK 文档

ConsoleAppender将日志信息输出到控制台标准输出流

```
<appender name="ConsoleAppender" type="log4net.Appender.ConsoleAppender" >
<layout type="log4net.Layout.PatternLayout">
<param name="ConversionPattern" value="%d [%t] %-5p %c [%x] - %m%n" />
</layout>
</appender>
```

EventLogAppender
详情参考 log4net.Appender.EventLogAppender SDK 文档

EventLogAppender将日志写入本地机器应用事件日志中默认情况下该日志源(Source)是AppDo.FriendlyName也可以手动指定其它名称

```
<appender name="EventLogAppender" type="log4net.Appender.EventLogAppender" >
<layout type="log4net.Layout.PatternLayout">
<param name="ConversionPattern" value="%d [%t] %-5p %c [%x] - %m%n" />
</layout>
</appender>
```

FileAppender

详情参考 log4net.Appender.File Appender SDK 文档

FileAppender将日志信息输出到指定日志文件

```
<!--[ !vml]-->

<appender name="LogFileAppender" type="log4net.Appender.FileAppender" >
```

<param name="File" value="WebUtilClient.log" />
<param name="AppendToFile" value="true" />
<layout type="log4net.Layout.PatternLayout">
<param name="ConversionPattern" value="%d [%t] %-5p %c [%x] - %m%n" />
</layout>
</appender>
File指定了文件名称可以使用相对路径此时日志文件位置取决于项目类型(如控制台、Windows Forms、ASP.NET等)；也可以使用绝对路径；甚至可以使用环境变量如<file value="${TMP}\log-file.txt" />
AppendToFile指定是追加到还是覆盖掉已有日志文件
还可以添加如下属性<lockingModel type="log4net.Appender.FileAppender+MinimalLock" />来使用最小锁定模型(minimal locking model)以允许多个进程可以写入同个文件
ForwardingAppender

详情参考 log4net.Appender.ForwardingAppender SDK 文档

ForwardingAppender可以用来为个Appender指定组约束看下面这个举例:

<appender name="ForwardingAppender" type="log4net.Appender.ForwardingAppender" >
<threshold value="WARN"/>
<appender-ref ref="ConsoleAppender" />
</appender>
在这个举例中为ConsoleAppender添加了约束Threshold为WARN这意味着对于条日志信息如果直接使用ConsoleAppender那么不论它是什么级别总会进行输出而如果使用这个ForwardingAppender则只有那些WARN或WARN以上日志才会发送到ConsoleAppender

MemoryAppender

详情参考 log4net.Appender.MemoryAppender SDK 文档

似乎不应该使用配置文件来配置MemoryAppender但如果你非要这么做看看这个举例(未验证):

<appender name="MemoryAppender" type="log4net.Appender.MemoryAppender">
<onlyFixPartialEventData value="true" />

</appender>
NetSendAppender
详情参考 log4net.Appender.NetSendAppender SDK 文档

NetSendAppender向特定用户屏幕发送消息(未验证)

```
<appender name="NetSendAppender" type="log4net.Appender.NetSendAppender">
<threshold value="ERROR" />
<server value="Anders" />
<recipient value="xym" />
<layout type="log4net.Layout.PatternLayout">
<conversionPattern value="%date [%thread] %-5level %logger [%property{NDC}] - %message%line"
/>
</layout>
</appender>
```

OutputDebugStringAppender

详情参考 log4net.Appender.OutputDebugStringAppender SDK 文档
下面这个例子描述了如何配置该Appender以向OutputDebugString API写入日志(未验证)

```
<appender name="OutputDebugStringAppender"
type="log4net.Appender.OutputDebugStringAppender" >
<layout type="log4net.Layout.PatternLayout">
<conversionPattern value="%date [%thread] %-5level %logger [%property{NDC}] - %message%line"
/>
</layout>
</appender>
```

RemotingAppender
详情参考 log4net.Appender.RemotingAppender SDK 文档

RemotingAppender向特定Sink发送日志信息(未验证):

<!--[!vml]-->

<appender name="RemotingAppender" type="log4net.Appender.RemotingAppender" >
<sink value="tcp://localhost:8085/LoggingSink" />
<lossy value="false" />
<bufferSize value="95" />
<onlyFixPartialEventData value="true" />
</appender>
RollingFileAppender

详情参考 log4net.Appender.RollingFileAppender SDK 文档

RollingFileAppender以FileAppender为基础和后者有着相同配置选项

下面这个例子演示了如何配置RollingFileAppender以写入log.txt文件写入文件名总是为log.txt(StaticLogFileName参数指定为true)；根据文件大小(RollingStyle)来生成新文件；最多保存有10个文件(MaxSizeRollBackups属性而且旦写满10个文件就不再写入日志了)每个文件最大为10KB这些文件名称为log.txt.1, log.txt.2...等

<appender name="RollingFileAppender" type="log4net.Appender.RollingFileAppender">
<param name="File" value="log\Log.txt" />
<param name="AppendToFile" value="true" />
<param name="MaxSizeRollBackups" value="10" />
<param name="MaximumFileSize" value="5MB" />
<param name="RollingStyle" value="Size" />
<param name="StaticLogFileName" value="true" />
<layout type="log4net.Layout.PatternLayout">
<param name="ConversionPattern" value="%d [%t] %-5p %c [%x] - %m%n" />
</layout>
</appender>
SmtpAppender

详情参考 log4net.Appender.SmtpAppender SDK 文档
SmtpAppender通过Smtp邮件服务器发送日志信息:

```
<appender name="SmtpAppender" type="log4net.Appender.SmtpAppender">
<authentication value="Basic" />
<to value="anderscui@tom.com" />
<from value="anderscui@163.com" />
<username value="anderscui" />
<password value="password" />
<subject value="test logging message" />
<smtpHost value="smtp.163.com" />
<bufferSize value="512" />
<lossy value="true" />
<evaluator type="log4net.Core.LevelEvaluator">
<threshold value="WARN"/>
</evaluator>
<layout type="log4net.Layout.PatternLayout">
<conversionPattern value="%line%date [%thread] %-5level %logger [%property{NDC}] -
%message%line%line%line" />
</layout>
</appender>
```

将其中to、from、username、password、subject、smtpHost配置正确才可能发送成功bufferSize可将多条信息打包在个邮件中evaluator可以对日志进行过滤

SmtpPickupDirAppender

详情参考 log4net.Appender.SmtpPickupDirAppender SDK 文档

配置和SmtpAppender类似但要把SmtpHost换为PickupDir(未验证)

```
<appender name="SmtpPickupDirAppender" type="log4net.Appender.SmtpPickupDirAppender">
<to value="to@do.com" />

<from value="from@do.com" />
<subject value="test logging message" />
<pickupDir value="C:\SmtpPickup" />
```

```
<bufferSize value="512" />
<lossy value="true" />
<evaluator type="log4net.Core.LevelEvaluator">
<threshold value="WARN"/>
</evaluator>
<layout type="log4net.Layout.PatternLayout">
<conversionPattern value="%line%date [%thread] %-5level %logger [%property{NDC}] -
%message%line%line%line" />
</layout>
</appender>
```
TraceAppender

详情参考 log4net.Appender.TraceAppender SDK 文档

TraceAppender将日志信息写入.Diagnostics.Trace系统(出现在输出窗口)

```
<appender name="TraceAppender" type="log4net.Appender.TraceAppender">
<layout type="log4net.Layout.PatternLayout">
<conversionPattern value="%date [%thread] %-5level %logger [%property{NDC}] - %message%line"
/>
</layout>
</appender>
```
UdpAppender
详情参考 log4net.Appender.UdpAppender SDK 文档

下例演示了如何配置UdpAppender(未验证):

```
<appender name="UdpAppender" type="log4net.Appender.UdpAppender">
<localPort value="8080" />
<remoteAddress value="224.0.0.1" />
<remotePort value="8080" />
<layout type="log4net.Layout.PatternLayout, log4net">
<conversionPattern value="%-5level %logger [%property{NDC}] - %message%line" />
</layout>
</appender>
```
上面有若干个Appender标注为"未验证"是指这些Appender极少用到或者在我机器上没能实现 :(

2009-2-12 5:09:54
疯狂代码 http://CrazyCoder.cn/