

## log4net的各种Appender配置示例

2007-03-09 17:02 by Anders Cui, 4194 visits, [收藏](#), [编辑](#)

在log4net的配置中，**appender**是最重要的部分，一般来说，每一种**appender**都表示一种日志的输出介质，如日志文件、**EvengLog**、数据库、控制台、邮件、**ASP.NET**页面等。

本文对各种内置的**appender**的配置提供了示例，但却远称不上详尽。要想了解每一种**appender**的参数和选项的说明，请参看该**appender**的SDK文档。

以下示例都是.NET 2.0下进行的，log4net的版本为1.2.10。

### AdoNetAppender

详情参考 [log4net.Appender.AdoNetAppender SDK 文档](#)。

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

首先建立数据表：

```
CREATE TABLE [dbo].[Log]
(
    [Id] [int] 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="System.Data.SqlClient.SqlConnection, System.Data, Version=2.0.0.0, Culture=Neutral, PublicKeyToken=b77a5c561934e089" />
  <connectionString value="server=(local);database=TestBase;integrated 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*表示批处理的日志事件，可以避免每次日志事件都访问数据库；*ConnectionType*指定了要使用的IDbConnection的完全限定类型名称；*connectionString*表示连接字符串；*CommandText*是SQL语句或存储过程；最后一组*parameter*节点描述了SQL语句或存储过程需要的参数。

## 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%newline"
/>
</layout>
</appender>
```

这段配置可将日志信息输出到页面的Trace上下文环境。如果日志的级别低于WARN，会以System.Web.TraceContext.Write方法输出；如果级别为WARN或WARN以上则会以System.Web.TraceContext.Warn方法输出，下图中的日志信息的不同颜色可以说明这一点。效果图如下：

aspx.page	Begin Load
logger	2007-03-09 13:07:34,125 [10] ERROR logger [(null)] - message text
logger	2007-03-09 13:07:34,171 [10] INFO logger [(null)] - message text
Default.aspx	2007-03-09 13:07:34,187 [10] ERROR Default.aspx [(null)] - message text
Default.aspx	2007-03-09 13:07:34,187 [10] INFO Default.aspx [(null)] - message text
aspx.page	End Load
aspx.page	Begin LoadComplete
aspx.page	End LoadComplete

这在进行页面调试的时候可是很方便的。

### 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的主要作用是将输出到指定类型（这里是LogFileAppender）的Appender的日志信息进行缓存。bufferSize属性指定了缓存的数量，如果value为5，那么将在信息量达到6条的时候，把这些日志批量输出。appender-ref属性指定了缓存的Appender类型，同root节点一样，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。下面这个示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>
</appender>
```

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

效果如下:

```
2007-03-09 13:56:37.265 [11] ERROR logger [<null>] - message text
2007-03-09 13:56:37.312 [11] INFO logger [<null>] - message text
2007-03-09 13:56:37.312 [11] ERROR UtilClient.Program [<null>] - message text
2007-03-09 13:56:37.328 [11] INFO UtilClient.Program [<null>] - message text
```

还可以为不同的级别指定不同的颜色:

```
<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%newline"
/>
  </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的应用程序事件日志中。默认情况下，该日志的源（Source）是AppDomain.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的日志文件。

```
<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，此时日志文件的位置取决于项目的类型（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的日志文件。

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一个文件。

ForwardingAppender

详情参考 [log4net.Appender.ForwardingAppender SDK 文档](#)。

ForwardingAppender可以用来为一个Appender指定一组约束。看下面这个示例：

```
<appender name="ForwardingAppender" type="log4net.Appender.ForwardingAppender" >
  <threshold value="WARN"/>
  <appender-ref ref="ConsoleAppender" />
</appender>
```

在这个示例中，为ConsoleAppender添加了约束，Threshold为WARN。这意味着对于一条日志信息，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以上的日志才会发送到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，看看这个示例（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向特定用户的屏幕发送消息（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%newline" />
  </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%newline" />
  </layout>
</appender>
```

RemotingAppender

详情参考 [log4net.Appender.RemotingAppender SDK 文档](#)。

RemotingAppender向特定的Sink发送日志信息（未验证）：

```
<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，与后者有着相同的配置选项。

下面这个例子演示了如何配置**RollingFileAppender**以写入**log.txt**文件。写入的文件名总是为**log.txt**（**StaticLogFileName**参数指定为**true**）；根据文件大小（**RollingStyle**）来生成新的文件；最多保存有**10**个文件（**MaxSizeRollBackups**属性，而且一旦写满**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="%newline%date [%thread] %-5level %logger [%property{NDC}] - %message%newline%newline%newline" />
  </layout>
</appender>
```

将其中的**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一个邮件中。**evaluator**可以对日志进行过滤。

**SmtpPickupDirAppender**

详情参考 [log4net.Appender.SmtpPickupDirAppender SDK 文档](#)。

配置与SmtpAppender类似，但要把SmtpHost换为PickupDir（未验证）。

```
<appender name="SmtpPickupDirAppender" type="log4net.Appender.SmtpPickupDirAppender">
  <to value="to@domain.com" />
  <from value="from@domain.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="%newline%date [%thread] %-5level %logger [%property{NDC}] - %message%newline%newline%newline" />
  </layout>
</appender>
```

TraceAppender

详情参考 [log4net.Appender.TraceAppender SDK 文档](#)。

TraceAppender将日志信息写入System.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%newline" />
  </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%newline" />
  </layout>
</appender>
```

上面有若干个Appender标注为"未验证"的，是指这些Appender极少用到，或者在我的机器上没能实现：(

希望这些内容能对您有所帮助。



By Anders Cui

原文地址：<http://logging.apache.org/log4net/release/config-examples.html>

作者：[Anders Cui](#)  
出处：<http://anderslly.cnblogs.com>  
本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，否则保留追究法律责任的权利。

 **Categories:** [45-Cool Tools](#)

 3 条回复

[ADD YOUR COMMENT](#) 

 **#1楼** [aimar](#) 

2007-09-23 23:43

请问如何配置smtp，对需要验证的邮箱地址如何处理，我将验证设置为Basic，可是仍然无法发送  
请问是怎么回事？  
谢谢！

[回复](#) [引用](#) [查看](#)

 **#2楼**[楼主] [Anders Cui](#) 

2007-09-24 10:30

@aimar  
那你看我的配置跟你的一样吗  
我当时的试验是成功的  
看了官方文档说,如果设置为Basic,那么用户名和密码一定要设置  
<http://logging.apache.org/log4net/release/sdk/log4net.Appender.SmtpAppender.html>

[回复](#) [引用](#) [查看](#)

 **#3楼** [一线风](#) 

2010-07-09 09:50

UdpAppender 这个很好用的哟，嘿嘿  
<http://wstask.codeplex.com/releases/view/47793>  
这个是我写的控制台监视器，如果也是Color的就好了。

[回复](#) [引用](#) [查看](#)

 注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[返回博客园首页](#)。

- 最新IT新闻:
- [英特尔强制要求接受资助的大学将研究成果开源](#)
  - [uTorrent官方软件被黑客替换成恶意程序](#)
  - [WiFi难民以美国无线电静默区为避难所](#)
  - [美国交友网站Match.com收购珍爱网20%股权](#)
  - [Monster让求职和招聘更容易？毛！](#)
- » [更多新闻...](#)

- 最新知识库文章:
- [利用Httponly提升web应用程序安全性](#)
  - [理解RESTful架构](#)
  - [我不是一个工程师 – 我是一个软件开发者](#)
  - [划分软件开发人员的两种尺度](#)
  - [为什么新人不会编程](#)
- » [更多知识库文章...](#)



China-pub 2011秋季教材巡展

China-Pub 计算机绝版图书按需印刷服务

网站导航: [网站首页](#) [知识库](#) [IT新闻](#) [我的园子](#) [闪存](#) [程序员招聘](#) [博问](#)

## About

程序员? 能体悟技术、全心投入、荣耀、快乐和团队精神  
旅行者? 希望能环游世界, 走遍读过的每一个国家和城镇



[我的豆瓣主页](#)

豆瓣douban.com

2008年12月 GDI (微软最有影响力开发者)  
2010年1月 C# MVP

昵称: [Anders Cui](#)  
园龄: [5年9个月](#)  
粉丝: [173](#)  
关注: [58](#)



SEARCH

最新评论

- [Re: Visual Studio 2008 可扩展性开发（七）：操作文本编辑器](#)
- @lexiaoyao20
- 这个功能在Visual Studio里面就有了啊:) -- Anders Cui
- [Re: Visual Studio 2008 可扩展性开发（七）：操作文本编辑器](#)
- 楼主，我想问一下，如果要分析代码，将类中没有使用到的命名空间删除掉，该如何做呢？ -- lexiaoyao20
- [Re: Visual Studio 2008 可扩展性开发（二）：Macro和Add-In初探](#)
- 不错，收获挺大的，非常感谢楼主。 -- lexiaoyao20
- [Re: DataGridView常见用法和FAQ汇总](#)
- 你好，我正在使用datagridview空间的rowRemove事件，但是我一运行软件时，即初始化Datagridview时。，自动运行rowRemove事件。还运行了7次。不知道是什么原因导致的。？ ... -- wtq
- [Re: 编写自文档化的代码](#)
- 可释义的代码。。。一直在努力。。。 -- Sylar.wang

日历

<2011年9月>

日	一	二	三	四	五	六
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

随笔档案

2011年6月(3)

2011年5月(1)

2011年3月(6)

2011年1月(1)

2010年1月(2)

2009年12月(1)

2009年8月(3)

2009年7月(4)

2009年6月(1)

2009年5月(9)

2009年4月(4)

2009年3月(8)

更多...

随笔分类

00-.NET Framework(18)

01-ASP.NET(16)

02-Winows Forms(10)

03-C# Recipes(2)

04-SQL Server(2)

05-XML(0)

06-AJAX(0)

07-JavaScript(7)

20-Agile Development(7)

21-Best Practice(1)

22-Design Patterns(4)

23-单元测试(5)

24-Open Source(14)

25-重构(2)

26-CodeSnippets(1)

27-iBATIS(17)

40-离散数学(5)

41-数据结构与算法分析(7)

42-Coding For Fun(24)

43-Python(1)

44-F#(29)

我管理的小组

Visual Studio

iBATIS.NET

单元测试

Scheme

SQLite

NBA

Haskell

我参加的小组

每日一句英语

jQuery

设计模式

博客园上海俱乐部

读书(Books)

python

书籍推荐

	<a href="#">45-Cool Tools(17)</a>	
	<a href="#">46-Visual Studio(18)</a>	
	<a href="#">47-VS Extensibility(15)</a>	
	<a href="#">60-随笔(29)</a>	
	<a href="#">61-业界新闻(7)</a>	
	<a href="#">62-英语学习(1)</a>	
	<a href="#">63-翻译(2)</a>	
	<a href="#">64-悦读(7)</a>	
	<a href="#">65-环境保护(1)</a>	

团队博客

	<a href="#">上海.NET俱乐部(0/0)</a>
	<a href="#">软件项目管理团队(0/0)</a>
	<a href="#">CLR基础研究团队(0/0)</a>
	<a href="#">框架设计团队(0/0)</a>
	<a href="#">F#开发(0/0)</a>

阅读排行榜

	<a href="#">DataGridView常见用法和FAQ汇总(21347)</a>
	<a href="#">理解.NET中的异常（二）(18990)</a>
	<a href="#">iBATIS In Action：什么是iBATIS（一）(10864)</a>
	<a href="#">iBATIS In Action：iBATIS的安装和配置(10578)</a>
	<a href="#">增强我们的Visual Studio（更新中）(9832)</a>
	<a href="#">推荐VS2008插件CodeRush Xpress for C#(9539)</a>
	<a href="#">iBATIS In Action：使用映射语句（一）(9163)</a>
	<a href="#">iBATIS In Action：使用高级查询技术（一）(8566)</a>
	<a href="#">使用iBatis和ObjectDataSource轻松实现分页(8088)</a>
	<a href="#">iBATIS In Action：什么是iBATIS（二）(7796)</a>

	<a href="#">IT文学</a>
	<a href="#">边走边拍</a>
	<a href="#">程序员音乐空间</a>



F#

	<a href="#">Don Syme</a>
	<a href="#">F# at Microsoft Research</a>
	<a href="#">F# Dev Center</a>
	<a href="#">lvxuwen</a>
	<a href="#">Robert Pickering</a>

FRIENDS

	<a href="#">Allen Lee</a>
	<a href="#">Angel Lucifer</a>
	<a href="#">Steven Pei</a>
	<a href="#">韦恩卑鄙</a>
	<a href="#">小力</a>
	<a href="#">有所为，有所不为</a>

OPEN SOURCE

	<a href="#">cs-open.com</a>
	<a href="#">sourceforge</a>

PORTAL SITE

	<a href="#">ASP.NET Official</a>
	<a href="#">C# Corner</a>
	<a href="#">CodeProject</a>
	<a href="#">WinForm Official</a>

PYTHON

	<a href="#">woodpecker</a>
---	----------------------------

大拿

	<a href="#">dflying chen</a>
	<a href="#">fantasySoft</a>
	<a href="#">idior</a>
	<a href="#">Jeffery Zhao</a>
	<a href="#">shanyou</a>
	<a href="#">Simon Duan</a>