编号：CCN-ZJ-7.3-02

团队开发框架规范与最佳实践

上海中商网络股份有限公司

2013年08月

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文档变更历史 | | | | |
| 版本号 | 日期（年/月/日） | 说明 | 编辑 | 审核 |
| V 1.0.0 | 2013/08/26 | 初稿 | 张宇 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[团队开发框架规范与最佳实践 1](#_Toc365310645)

[1 日志记录框架 4](#_Toc365310646)

[1.1 概念 4](#_Toc365310647)

[1.2 目的 4](#_Toc365310648)

[1.3 使用原则 4](#_Toc365310649)

[1.4 使用组件 4](#_Toc365310650)

[1.5 工具介绍以及使用 5](#_Toc365310651)

[1.5.1 企业库日志模块介绍 5](#_Toc365310652)

[1.5.2 企业库日志应用程序模块工作原理图: 6](#_Toc365310653)

[1.5.3 企业库日志模块特性 7](#_Toc365310654)

[1.5.4 企业库日志模块常用属性以及设置 7](#_Toc365310655)

[1.5.4.1 日志项概述 7](#_Toc365310656)

[1.5.4.2 几个重要属性及概念 8](#_Toc365310657)

[1.5.5 快速使用 8](#_Toc365310658)

[1.5.6 特性介绍以及使用(参考文献: http://wenku.it168.com/d\_000930540.shtml) 8](#_Toc365310659)

[1.5.6.1 在日志里面包含名-值的字典: 9](#_Toc365310660)

[1.5.6.2 跟踪活动并记录上下文信息 10](#_Toc365310661)

[2 异常管理框架 11](#_Toc365310662)

[3 验证规则框架 11](#_Toc365310663)

[4 数据访问框架 11](#_Toc365310664)

# 日志记录框架

## 概念

开发人员经常需要为应用程序编写日志记录功能。典型的，响应应用事件的格式日志信息。例如,他们可能需要记录一些非预期的日志,比如应用程序异常,或未能连接到一个数据库。开发者还需要写代码来跟踪应用程序流程以及某个组件在一个应用程序运行时用例或场景。此外,应用程序通常需要将日志写在本地和网络。在某些情况下,您可能需要将多个事件来源整理到一个单独的位置。

在一个完整的信息系统里面，日志系统是一个非常重要的功能组成部分。它可以记录下系统所产生的所有行为，并按照某种规范表达出来。我们可以使用日志系统所记录的信息为系统进行排错，优化系统的性能，或者根据这些信息调整系统的行为.

在某些情况，日志模块非常有用，例如，你需要在优先级和目录上过滤日志，你需要格式化日志信息，你需要在不修改代码的情况下变更日志的记录位置。日志模块设计的很灵活，可以自定义格式和trace listener，满足你的特定需要。

## 目的

* + 在应用程序和整个企业中维护一致的日志和规范做法.
  + 监视代码中的变量的变化情况，把数据周期性地记录到文件中供其它应用进行统计分析工作
  + 跟踪代码运行的轨迹，作为日后审计的依据
  + 担当集成开发环境中的调试器，向文件或者控制台打印代码的调试信息

## 使用原则

* + 用代码增加和引发事件
  + 增加带有其他环境信息的日志消息
  + 跟踪应用的行为，增加上下文信息
  + 构造日志信息之前，检查filter的状态

## 使用组件

日志记录组件采用微软企业库中的**Logging Application Block,**  开发人员可以使用此模块来处理系统中的不同的日志需求。

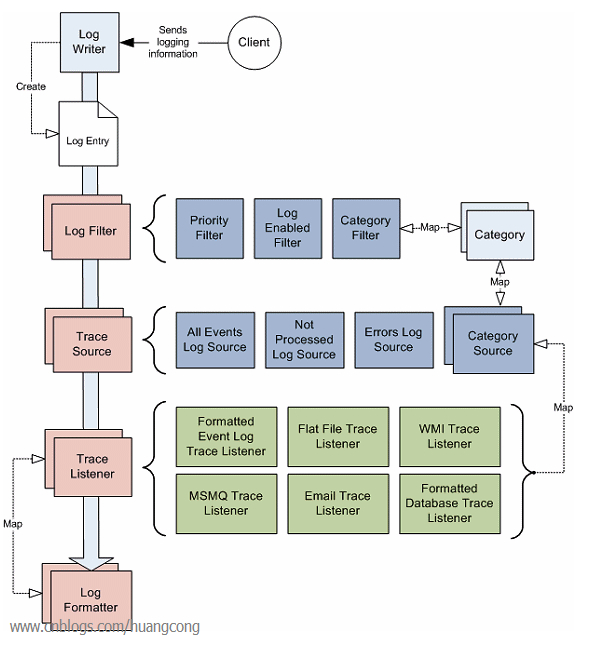
## 工具介绍以及使用

### 企业库日志模块介绍

微软企业库的日志模块实现了日志以及监控功能。开发者可以利用日志和监控模块在多个位置记录事件：

* + 事件日志
  + 电子邮件
  + 数据库
  + 消息队列
  + 文本文件
  + WMI的事件查看器
  + 自定义的位置

### 企业库日志[应用程序](http://www.haogongju.net/tag/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F" \o "应用程序" \t "_blank)模块工作原理图**:**



从上图我们可以看清楚企业库日志应用程序模块的工作原理,主要分为以下四大模块:

* 1. **Log Writer**: 日志写入器作为主要的入口点用来创建日志内容并将其写入到所选择的日志类型.
  2. **Log Filters:** 日志过滤器可以阻止或容许一个基于数量特性的日志条目.每个日志对象可以分配一个或多个类别(默认为一般类别),类别日志过滤器可以使用类别来过滤日志.此外,有两个特殊的日志过滤器可以组织所有的日志,或者阻止日志级别定于设定值.所有的这些可以在配置文件中定义,类别,重要性以及日志过滤器
  3. **Trace Sources**:跟踪源
  4. **Log Formatters**:你可以针对每个跟踪点添加格式化配置,程序会将记录的日志格式化为你需要的日志格式.模块包含一个文本格式化工具,您可以配置与跟踪侦听器,将日志转化为您所需要的格式,如磁盘文件,电子邮件,windows事件.同时包含一个二进制格式化工具,将其序列化适合传输的目标,如序列化日志到windows消息队列(MSMQ).文本格式化程序是可配置的,您可以自定义过文本格式和文本信息,包括使用占位符.

其中LogFilter,Trace Source,Trace Listener,Log Formatter的信息都可以在Category配置文件中反映出来,通过配置文件,调用LogWriter类的Writer方法,就可以将包含日志信息的LogEntry实体写入Category配置文件中指定的设备了.

### 企业库日志模块特性

* 高可配置特性:可配置过滤器,定制消息格式,自定义Sink,formatter等
* 强大的可视化配置工具,配置方便
* 支持多方式存储日志,如:文件,邮件,数据库,xml,mssq
* 会随着企业库的升级而进行版本更新.

### 企业库日志模块常用属性以及设置

#### 日志项概述

在记录日志时,都是创建一个日志项来承载记录的信息,每个日志项具有以下属性:

* Message:消息内容(必须项,消息内容)
* Cagegory:消息类别(提供缺省值)
* Priority:日志优先级(缺省值-1)
* EventID:事件ID,(缺省值为-1)
* Severity:严重程度(缺省值为Severity Unsepcialfied)
* Title:标题(缺省值为””)

#### 几个重要属性及概念

Sink: 日志记录的位置

Category: 决定了你在程序中添加的日志写向何处,是通过配置来实现的.比如说我们有两个Category,第一个我们指定Sink为事件日志,第二个我们可以指定sink为文本文件.

Formatter: 格式器决定了日志记录的格式,通过配置实现.

### 快速使用

开发环境:VS2012+EntLib6+.NET4.5 (注:EntLib6必须要.NET 4.5支持)

1. 安装Entlib6
2. 新建控制台程序,添加对Microsoft.Practices.EnterpriseLibrary.Logging.dll 引用
3. 打开企业库界面配置工具EntLibConfig(默认在安装目录下)
4. 菜单选择Blocks->Add Logging Setting,不做任何设置,保存到控制台目录下.
5. 编写代码如下:

//设置日志写入器

Logger.SetLogWriter(new LogWriterFactory().Create());

//实例化日志对象

LogEntry logEntry = new LogEntry();

//设置事件ID

logEntry.EventId = 100;

//设置日志对象优先级

logEntry.Priority = 2;

//设置日志内容

logEntry.Message = "EntLib Test Informational message";

//设置目录类别

logEntry.Categories.Add("General");

//写入

Logger.Write(logEntry);

1. 执行完成后在windows日志查看器->应用程序日志,可以查看到刚才写入的日志.

### 特性介绍以及使用(参考文献: http://wenku.it168.com/d\_000930540.shtml)

#### 在日志里面包含名-值的字典:

在日志消息里,我们可以包含名-值这样的字典.这时需要创建一个HashTable,通过key-Value的方式把要记录的内容传入,实例代码如下:

//定义字典表

Dictionary<string, object> logTable = new Dictionary<string, object>(3);

logTable["Key\_1"] = "Value\_1";

logTable["Key\_2"] = "Value\_2";

logTable["Key\_3"] = "Value3";

//实力化日志对象

LogEntry logEntry = new LogEntry();

//设置属性值

logEntry.Message = "键值写入测试!";

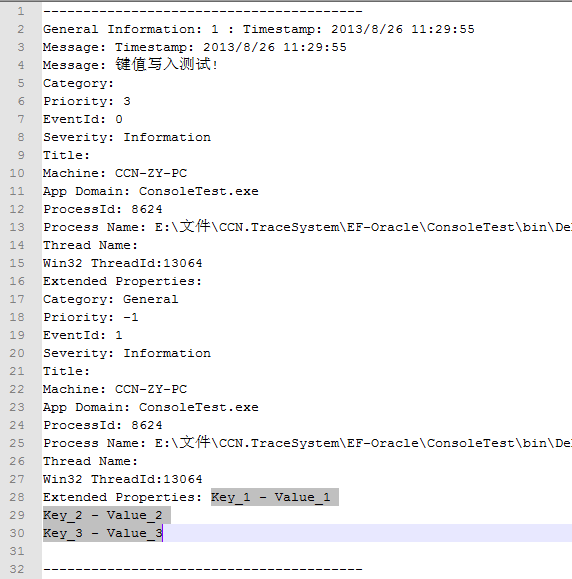
logEntry.Priority = 3;

logEntry.Categories.Add("General");

//写入日志

Logger.Write(logEntry,logTable);

查看写入的日志:



#### 跟踪活动并记录上下文信息

跟踪应用程序的活动是一项常见的功能,我们需要把活动的起始时间和结束时间以及活动的其他信息记录下来.日志和监控应用程序支持活动ID来跟踪一项活动,活动ID可以在代码中指定,也可以通过程序来自动生成,程序块自动记录活动的起始时间和结束时间.活动跟踪有Trace类来提供,示例代码如下:

# 异常管理框架

# 验证规则框架

# 数据访问框架