

# HashLog v. 1.0 User Documentation

---

END USER DOCUMENTATION

Piotr Paszko

# Table of Contents

## Table of Contents

.....	0
Table of Contents .....	1
Project Overview .....	2
License .....	3
Requirements .....	4
Installation.....	5
Using Example .....	6
API Documentation .....	7

# Project Overview

HashLog is simple and user friendly library for C# .Net Framework 4.6.1. Goal was simple: create most easy to use, runtime logging tool.

Library gets log messages from user and save it to external file and write in console. Everything is fully automatic.

# License

Project is licensed by GNU v. 3 license. Feel free to use and modify! For license conditions see **LICENSE** file.

# Requirements

Project is written for .NET Framework 4.6.1. Using it on other versions can make some problems.

# Installation

To set up HashLog in your project follow this steps:

1. Open HashLog.sln in Visual Studio.
2. Build HashLog project with your settings.
3. Create your own project.
4. Right click on your project in Visual Studio and click **Add->Reference**.
5. In pop up window click **Browse** button.
6. Select builded .dll file.
7. Check HashLog reference.
8. Click **OK** button.
9. Library ready to use!

# Using Example

Here's using example of HashLog library. Simple! Isn't it?

```
namespace HashLogTester
{
    class Program
    {
        static void Main(string[] args)
        {
            HashLog.HashLog.Setup("Test project");
            HashLog.HashLog.LogInformation("Info.");
            HashLog.HashLog.LogWarning("Warning.");
            HashLog.HashLog.LogError("Error.");

            HashLog.HashLog.Setup("Test project 2");
            HashLog.HashLog.LogFatalError("Fatal error");
        }
    }
}
```

This example will create two log files. One for each HashLog.HashLog.Setup() calls. That method should be called once when our application is starting. "Test project" is name of our project. Every other line will log our messages with suitable prefixes. But HashLog.HashLog.LogFatalError() will also close our application due fatal error.

# API Documentation

## class HashLog

Public Methods:

Name	Arguments	Description
<code>static void Setup(string projectName)</code>	<code>string</code> projectName - name of project used in log file and console title.	Method is doing setup of logger. Prepare log file etc.
<code>static void LogInformation(string message)</code>	<code>string</code> message - log message.	Logs message with information prefix. Use when logging actual program state.
<code>static void LogWarning(string message)</code>	<code>string</code> message - log message.	Logs message with warning prefix. Use when logging unexpected program problems only little affecting on operations.
<code>static void LogError(string message)</code>	<code>string</code> message - log message.	Logs message with error prefix. Use when logging unexpected program problems affecting on operations, for example breaking it.
<code>static void LogFatalError(string message)</code>	<code>string</code> message - log message.	Logs message with fatal error prefix and close application. Use when logging unexpected program problems affecting on core components making whole app unable to working.