Loxodon Framework NLog

license MIT
release v2.6.2

openupm v2.6.2

npm v2.6.2

Developed by Clark

Requires Unity 2018.4 or higher.

This plug-in integrates NLog into Loxodon.Framework. It is recommended to use this plug-in instead of the Log4Net plug-in. It allocates less heap memory during the log printing process.

For tutorials,examples and support,please see the project.You can also discuss the project in the Unity Forums.

The version is compatible with MacOSX,Windows,Linux,IOS and Android etc.

Installation

Install via OpenUPM (recommended)

OpenUPM can automatically manage dependencies, it is recommended to use it to install the framework.

Requires nodejs's npm and openupm-cli, if not installed please install them first.

```
# Install openupm-cli,please ignore if it is already installed.
npm install -g openupm-cli

#Go to the root directory of your project
cd F:/workspace/New Unity Project

#Install loxodon-framework-nlog
openupm add com.vovgou.loxodon-framework-nlog
```

Install via Packages/manifest.json

Modify the Packages/manifest.json file in your project, add the third-party repository "package.openupm.com"'s configuration and add "com.vovgou.loxodon-framework-log4net" in the "dependencies" node.

Installing the framework in this way does not require nodejs and openm-cli.

```
{
  "dependencies": {
    ...
    "com.unity.modules.xr": "1.0.0",
    "com.vovgou.loxodon-framework-nlog": "2.6.1"
  },
  "scopedRegistries": [
    {
      "name": "package.openupm.com",
      "url": "https://package.openupm.com",
      "scopes": [
        "com.vovgou",
        "com.openupm"
      ]
    }
  ]
}
```

Quick Start

If you are compiling with IL2CPP, you need to add the following configuration to the link.xml file to avoid NLog classes being stripped:

```
<linker>
    <assembly fullname="NLog">
       <namespace fullname="NLog" preserve="all"/>
    </assembly>
</linker>
```

Supported environment variables:

- persistent-data-path : UnityEngine.Application.persistentDataPath
- temporary-cache-path : UnityEngine.Application.temporaryCachePath

The NLog configuration file is as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <targets async="true">
        <target name="logfile" xsi:type="File"
                                bufferSize="8192"
                                openFileCacheTimeout="30"
                                keepFileOpen="false"
                                deleteOldFileOnStartup="false"
                                fileName="${persistent-data-path}/logs/${shortdate}.log"
                                layout="${longdate} [${uppercase:${level}}] ${callsite}(${callsi
                <target name="logconsole" xsi:type="UnityConsole"
                                layout="${longdate} [${uppercase:${level}}] ${callsite}(${callsi
        </targets>
    <rules>
        <logger name="*" minlevel="Debug" writeTo="logconsole" />
        <logger name="*" minlevel="Debug" writeTo="logfile" />
    </rules>
</nlog>
```

Read NLog configuration file and create LogFactory.

```
public class NLogManager : MonoBehaviour
{
    void Awake()
    {
        ////Load the NLog configuration file from the StreamingAssets directory
        //Loxodon.Log.LogManager.Registry(NLogFactory.Load(Application.streamingAssetsPath + "/c

        //Load the NLog configuration file from the Resources directory
        Loxodon.Log.LogManager.Registry(NLogFactory.LoadInResources("config"));

        DontDestroyOnLoad(this.gameObject);
    }

    void OnDestroy()
    {
        NLogFactory.Shutdown();
    }
}
```

Usage example.

```
public class NLogExample : MonoBehaviour
{
    private ILog log;
    void Start()
    {
        log = LogManager.GetLogger(typeof(NLogExample));

        log.Debug("This is a debug test.");
    }
}
```

Contact Us

Email: yangpc.china@gmail.com

Website: https://vovgou.github.io/loxodon-framework/

QQ Group: 622321589

加入QQ群