

Quick start guide

Custom Logging via Log4Net Library

SER0012 - Log4Net - iX Sample



1 Function and area of use

This document provides guidelines when working with Sample_Log4Net.

This document explains how to enable, disable and copy log file to USB via iX actions. It also explains how to log your own debug messages. Logging custom debug messages will help you to debug an iX application in different production lifetime.

For example, you can log the DEBUG level message, and then the log can be used for checking the program scenarios as you expected in the development phase.

And logging ERROR level message in the production phase (in the field). It makes it possible to get back some information from your customers or different departments to analyze what might be the root cause of the system crash when they are using your iX applications.

The default level is INFO in iX automatically generated log4net.config. You can modify it before downloading the iX application to an X2 or iX RT target. For the log level, please refer to Apache log4net project official website.

Apache log4net™

The Apache log4net library is a tool to help the programmer output log statements to a variety of output targets.

Apache log4net™ Manual - Introduction
<https://logging.apache.org/log4net/release/manual/introduction.html>

Debug Logging (System Tag): True

C:\Users\yexhu\Documents\iX Developer 2.40 SP4 Projects\Sample_Log4Net\Temp\Output

名稱	修改日期	類型	大小
Ionic.Zip.dll	2019/10/4 上午 10:29	應用程式擴充	452 KB
LicenseResolverTools.dll	2019/10/4 上午 11:40	應用程式擴充	17 KB
Log.log	2020/4/23 上午 12:15	文字文件	3 KB
Log.log.1	2020/4/23 上午 12:15	1 檔案	101 KB
log4net.config	2019/10/4 上午 10:30	XML Configurati...	3 KB
log4net.dll	2019/10/4 上午 10:30	應用程式擴充	192 KB
LumenWorks.Framework.IO.dll	2019/10/4 上午 10:29	應用程式擴充	35 KB
M2Mqtt.NetCF35.dll	2019/10/4 上午 06:36	應用程式擴充	53 KB
MacroExtensionsCF.dll	2019/10/4 上午 11:48	應用程式擴充	22 KB
MediaPlayerCELib.dll	2019/10/4 上午 10:29	應用程式擴充	8 KB

Log4Net Actions of iX Developer

Enable Debug Log

Disable Debug Log

Copy Debug Log to USB

Try it!
 You can find the logs within the Log.log file in the project folder.
 The old logs will be back up into Log.log.* files.

0

÷

0

=

0

Calculation

2 About this document

This quick start document should not be considered as a complete manual. It is an aid to be able to startup a normal application quickly and easily.

Copyright © Beijer Electronics, 2020

This documentation (below referred to as 'the material') is the property of Beijer Electronics. The holder or user has a non-exclusive right to use the material. The holder is not allowed to distribute the material to anyone outside his/ her organization except in cases where the material is part of a system that is supplied by the holder to his/ her customer. The material may only be used with products or software supplied by Beijer Electronics. Beijer Electronics assumes no responsibility for any defects in the material, or for any consequences that might arise from the use of the material. It is the responsibility of the holder to ensure that any systems, for whatever applications, which is based on or includes the material (whether in its entirety or in parts), meets the expected properties or functional requirements. Beijer Electronics has no obligation to supply the holder with updated versions.

Use the following hardware, software, drivers and utilities in order to obtain a stable application:

In this document we have used following software and hardware

- iX Developer 2.4 SP4 /SP5
- X2 Series and iX Runtime, including C2 Series.

For further information refer to

- [Apache log4net™ Manual - Introduction](#)
- [Beijer Electronics knowledge database, HelpOnline](#)

This document and other quick start documents can be obtained from our homepage. Please use the address support.europe@beijerelectronics.com for feedback.

3 Table of Contents

- 1 Function and area of use.....2
- 2 About this document.....3
- 3 Table of Contents.....4
- 4 Actions of the iX Developer5
 - 4.1 Actions5
 - 4.2 The same actions on X2 series service menu5
- 5 System Tag - Debug Logging Enabled6
- 6 Custom Log message7
 - 6.1 The default logging level of iX7
 - 6.2 Used in developing a custom assembly (DLL).9
- 7 Pitfalls 10
 - 7.1 The Rolling File Appender is used..... 10
- 8 About Beijer Electronics 11
 - 8.1 Contact us 11
 - Global offices and distributors 11

4 Actions of the iX Developer

iX Actions are good for a developer to develop an application to achieve some functions in a short time.

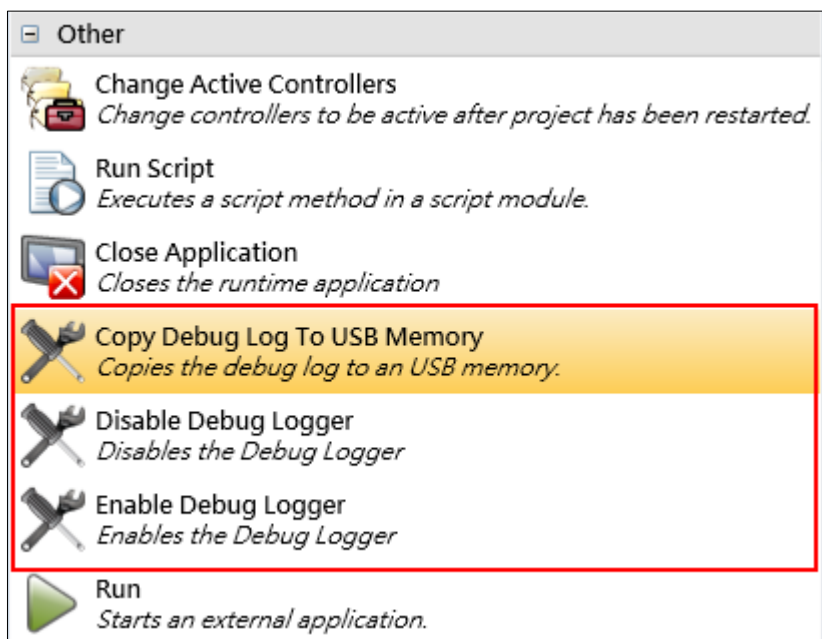
There are three actions in iX Developer belong to the Debug Logging function.

- Enable Debug Logger
- Disable Debug Logger
- Copy Debug Log to USB memory

Notes: It will directly copy the log to USB storage and delete the log that has been copied in the X2 series. In iX Runtime, it will let you browse the directory you want to save the copy.

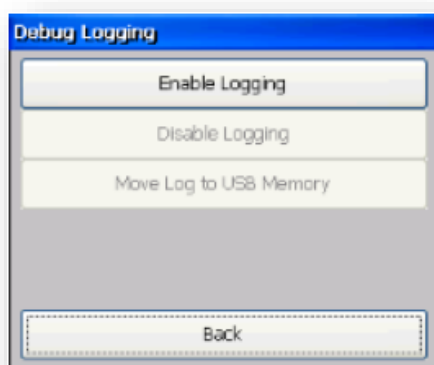
4.1 Actions

The Debug Logging actions are in the Other classification.



4.2 The same actions on X2 series service menu

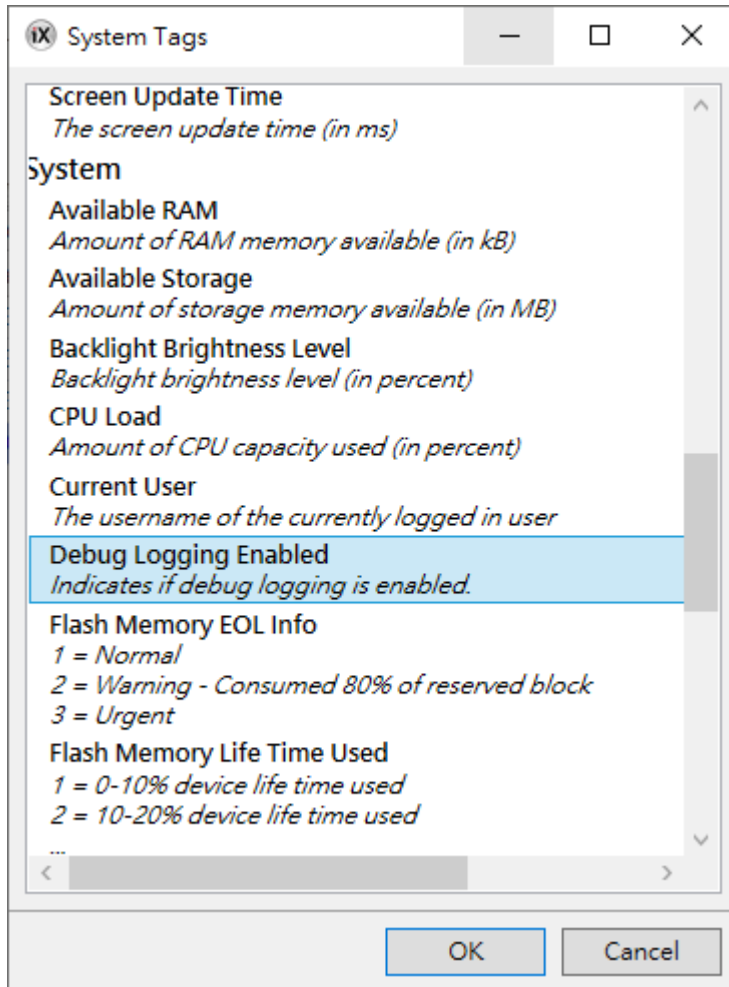
The same actions can be found on the service menu of an X2 panel. Therefore, you can directly call these actions before an iX application is running.



5 System Tag - Debug Logging Enabled

A System Tag can help you to get the system info in a quick way.

For Debug Logging, there is one built-in system tag, Debug Logging Enabled (SystemTagDebugLoggingEnabled). The enable/disable status of the Debug Logging can be gotten via this system tag. It indicates if the Debug Logging is enabled (Boolean: True/False).



SystemTagDebugLoggingEnabled ...	DEFAULT	Read	BOOL	Indicates if debug logging is enabled.
----------------------------------	---------	------	------	--

6 Custom Log message

Besides the iX system log message, you can log your own message in C# scripting.

In any class (Screen, Tags, AlarmServer, self-defined class, etc.), the logger can be gotten via the LogManager, and then you can log custom message at any place in the class.

```
public class DemoClass
{
    private static readonly log4net.ILog _log = log4net.LogManager.GetLogger(typeof(DemoClass));

    public static string DoSomething()
    {
        try
        {
            // do something and might go wrong
            _log.Debug("Try to do something")
        }
        catch(Exception ex)
        {
            _log.Error(ex.ToString());
        }
    }
}
```

In the real world, there are some unpredictable situations happened. So a Try/Catch statement can help us to catch the exceptions and handle them. Logging is one of the usual handle methods.

In this sample, it catches the DivideByZeroException to show you how it works like the above sample code. It also demonstrates it can log different level messages at any place as you desired.

6.1 The default logging level of iX

The default logging level is INFO, and it would be automatically generated in the configuration file, log4net.config, for the first time Build the project and every time Rebuild project. Therefore, before your application is running, you should change the level as your desired.

The configuration file is under the project folder:

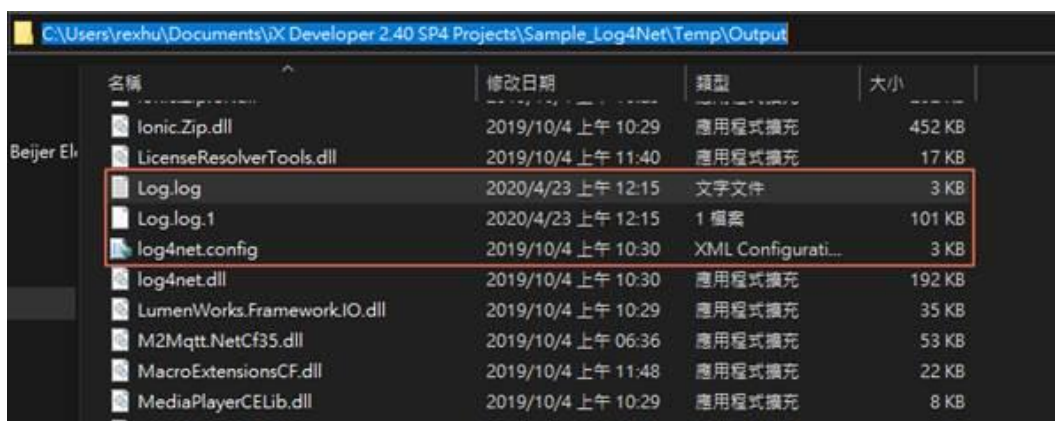
`{Project Folder}\Temp\Output\log4net.config`

e.g. Sample_Log4Net\Temp\Output\log4net.config

The place of the logging level setting you would like to change:

`<level value="INFO" />` in the log4net.config

For other settings, please refer to Apache log4net project official website.



In real world, it is very useful for software development. For example, you want to log DEBUG level message in development phase only. A programmer sometimes would like to log an unnecessary message by using the DEBUG level for his/her own development verification. So, the programmer can change the logging level back to INFO before the application is deployed in the field or delivered to the end-user without changing his/her codes.

[Default settings in log4net.config]

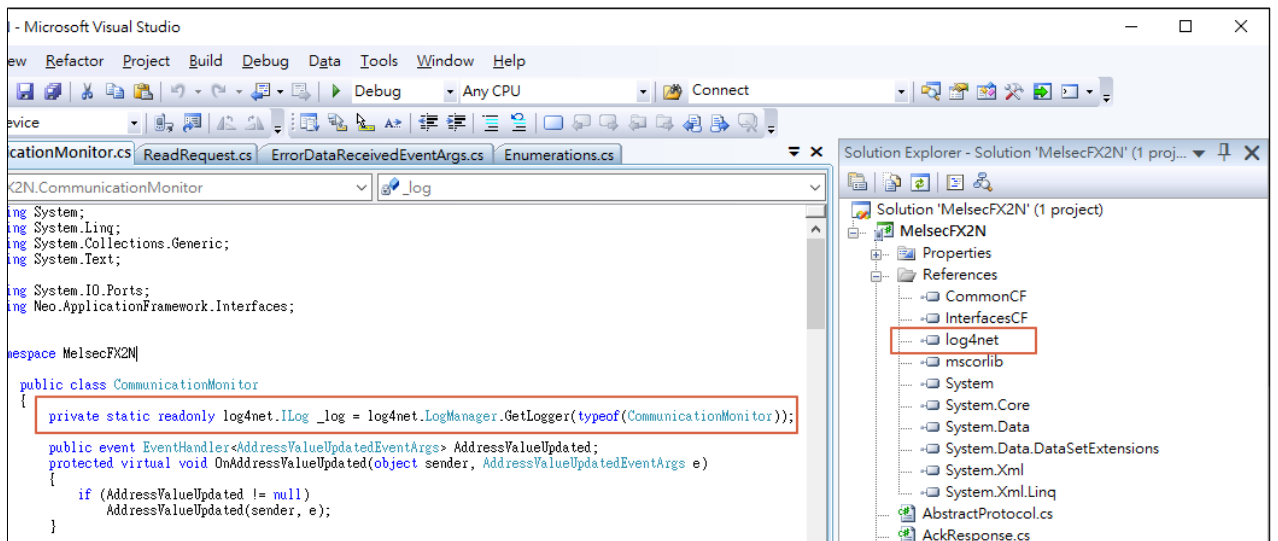
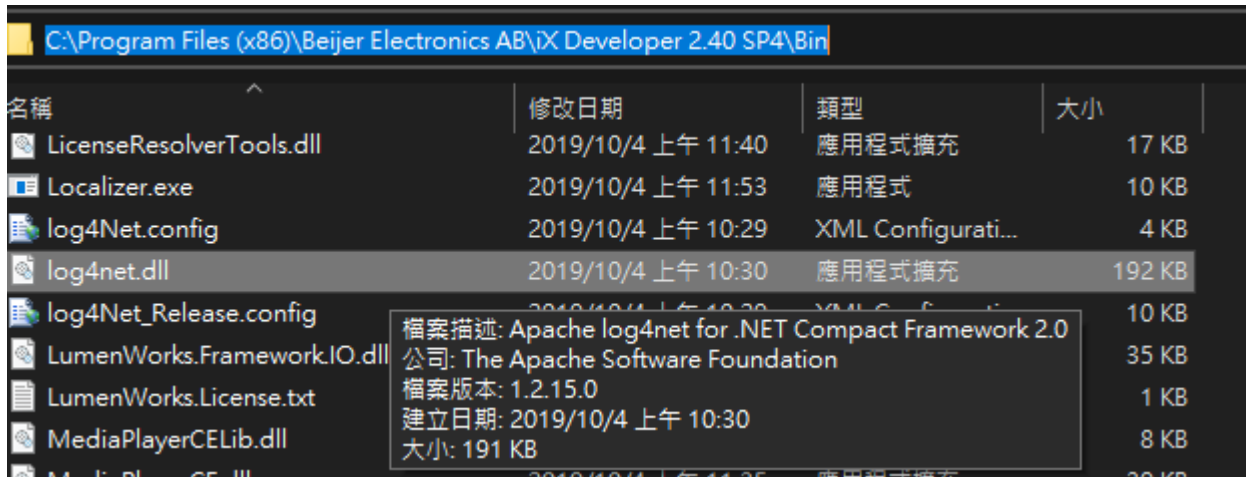
```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <!-- Register a section handler for the log4net section -->
  <configSections>
    <section name="log4net" type="System.Configuration.IgnoreSectionHandler" />
  </configSections>
  <appSettings>
    <!-- To enable internal log4net logging specify the following appSettings key -->
    <!-- <add key="log4net.Internal.Debug" value="true"/> -->
  </appSettings>
  <!-- This section contains the log4net configuration settings -->
  <log4net>
    <!-- Define some output appenders -->
    <appender name="RollingLogFileAppender" type="log4net.Appender.RollingFileAppender">
      <file value="Log.log" />
      <appendToFile value="true" />
      <maxSizeRollBackups value="10" />
      <maximumFileSize value="100KB" />
      <rollingStyle value="Size" />
      <staticLogFileName value="true" />
      <layout type="log4net.Layout.PatternLayout">
        <header type="log4net.Util.PatternString">
          <conversionPattern value="[Header %property{windowsversion}]%newline" />
        </header>
        <footer value="[Footer]
#10;" />
        <conversionPattern value="%date [%thread] %-5level %logger - %message%newline" />
      </layout>
    </appender>
    <appender name="TraceAppender" type="log4net.Appender.TraceAppender">
      <layout type="log4net.Layout.PatternLayout">
        <conversionPattern value="%date [%thread] %-5level %logger [%property{NDC}] - %message%newline" />
      </layout>
    </appender>
    <appender name="UdpAppender" type="log4net.Appender.UdpAppender">
      <param name="RemoteAddress" value="127.0.0.1" />
      <param name="RemotePort" value="8080" />
      <layout type="log4net.Layout.XmlLayout">
        <param name="Prefix" value="" />
      </layout>
    </appender>
    <!-- Setup the root category, add the appenders and set the default level -->
    <root>
      <level value="INFO" />
      <appender-ref ref="RollingLogFileAppender" />
      <appender-ref ref="TraceAppender" />
    </root>
  </log4net>
</configuration>
```


6.2 Used in developing a custom assembly (DLL).

When you are developing a library in Visual Studio, you can reference the log4net.dll in the iX installation folder.

{iX Developer installation folder}\Bin\ log4net.dll

e.g. C:\Program Files (x86)\Beijer Electronics AB\iX Developer 2.40 SP4\Bin\log4net.dll



7 Pitfalls

7.1 The Rolling File Appender is used

Logging in iX uses text files to save the log messages. File I/O is required. Logging should not be inserted in a loop statement. Otherwise, it might lead the application blocked until exiting the loop.

8 About Beijer Electronics

Beijer Electronics is a multinational, cross-industry innovator that connects people and technologies to optimize processes for business-critical applications. Our offer includes operator communication, automation solutions, digitalization, display solutions and support. As experts in user-friendly software, hardware

and services for the Industrial Internet of Things, we empower you to meet your challenges through leading-edge solutions.

Beijer Electronics is a Beijer Group company. Beijer Group has a sale over 1.6 billion SEK in 2019 and is listed on the NASDAQ OMX Nordic Stockholm Small Cap list under the ticker BELE. www.beijergroup.com

China

Shanghai

NORWAY

Drammen

TAIWAN

Taipei

DENMARK

Roskilde

SOUTH KOREA

Seoul

TURKEY

Istanbul

FRANCE

Paris

SWEDENGöteborg
Malmö
Stockholm**UNITED KINGDOM**

Nottingham

GERMANY

Nürtingen

USA

Salt Lake City

8.1 Contact us

[Global offices and distributors](#)