

NLog

This tutorial demonstrates how to implement [NLog](#) with the Coralogix platform.

Installation

To install Coralogix.NLog, [run the following command](#) in the Package Manager Console.

For **.NET 5+**, use:

```
Install-Package CoralogixCoreNLog
```

Configuration

Variables

Provide the following variables.

Variable	Description
Private Key	Your Coralogix Send-Your-Data API key
Application Name	The name of your application , as it will appear in your Coralogix dashboard. For example, a company named SuperData might insert the SuperData string parameter. If SuperData wants to debug its test environment, it might use SuperData-Test.
Subsystem Name	The name of your subsystem , as it will appear in your Coralogix dashboard. Applications often have multiple subsystems (ie. Backend Servers, Middleware, Frontend Servers, etc.). In order to help you examine the data you need, inserting the subsystem parameter is vital.

Via XML NLog Configuration

To use XML configuration, you must set the CORALOGIX_LOG_URL environment variable to the Coralogix [Logs endpoint](#) associated with your Coralogix [domain](#).

Example: **`https://ingress.eu2.coralogix.com:443/api/v1/logs`**

```
<?xml version="1.0" ?>
<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <extensions>
    <add assembly="CoralogixNLog"/>
  </extensions>
</nlog>
```

```

</extensions>
<targets>
  <target name="console" xsi:type="CoralogixNLog"
    PrivateKey="<Private Key>"
    ApplicationName="MyTestApp"
    SubsystemName="BL"
    layout="${date:format=HH\\\:MM\\\:ss} ${logger} ${message}" />
</targets>

<rules>
  <logger name="*" minlevel="Info" writeTo="console" />
</rules>
</nlog>

```

Via Code

```

using NLog;
using NLog.Config;
using NLog.Coralogix;

//Set to the ingress endpoint for your Coralogix Domain
Environment.SetEnvironmentVariable("CORALOGIX_LOG_URL", "
<https://ingress.eu2.coralogix.com>");

LoggingConfiguration config = new LoggingConfiguration();

CoralogixTarget coralogixTarget = new CoralogixTarget();
config.AddTarget("Coralogix", coralogixTarget);

//Configure the Coralogix target with the Coralogix settings.
//You need to configure this only once.
coralogixTarget.PrivateKey = "<Private Key>";
coralogixTarget.ApplicationName = "DotNet";
coralogixTarget.SubsystemName = "NLog";

coralogixTarget.Layout = @"${date:format=HH\\\:mm\\\:ss} ${logger}
${message}";

//Configure the Level filter for the Coralogix Target
var rule1 = new LoggingRule("*", NLog.LogLevel.Debug, coralogixTarget);
config.LoggingRules.Add(rule1);

```

```
// Define the actual NLog logger which through it all log entries should be reported
NLog.LogManager.Configuration = config;

// The common practice is to get an instance of the logger in each class
and setting the logger name to the class name.
//logger name will be used as category unless specified otherwise.
NLog.Logger nlogger = NLog.LogManager.GetLogger("MyApp");

nlogger.Debug("Debug Message");
nlogger.Info("Info Message");
nlogger.Warn("Warning Message");
nlogger.Error("Error Message");
nlogger.Trace("Trace Message");

Thread.Sleep(10000);

Console.WriteLine("Completed.");
```

Notes:

- You can also use the CORALOGIX_LOG_URL environment variable to set your endpoint instead of doing it in code if needed.
- To do so, set the CORALOGIX_LOG_URL environment variable to the Coralogix [Bulk Logs endpoint](#) associated with your Coralogix [domain](#).

Support

Need help?

Our world-class customer success team is available 24/7 to walk you through your setup and answer any questions that may come up.

Feel free to reach out to us **via our in-app chat** or by sending us an email at support@coralogix.com.

 Last updated: July 23, 2025

Was this helpful?

Leave your feedback [here](#).

☐ **Yes** ☐ **No**

Send

© 2025 Coralogix. All rights reserved.

Generated on: November 17, 2025

Source: <https://coralogix.com/docs/integrations/sdks/nlog/>