

 eBook Gratuit

# APPRENEZ

---

# nlog

eBook gratuit non affilié créé à partir des  
**contributeurs de Stack Overflow.**

#nlog

# Table des matières

<b>À propos</b> .....	<b>1</b>
<b>Chapitre 1: Démarrer avec nlog</b> .....	<b>2</b>
Remarques.....	2
Versions.....	2
Exemples.....	2
Écrire les premiers journaux.....	2
<b>Chapitre 2: Dépannage de NLog</b> .....	<b>5</b>
Exemples.....	5
Activer le journal interne dans nlog.config.....	5
Activer l'enregistreur interne par programmation.....	5
<b>Chapitre 3: Ecrire des journaux dans des fichiers</b> .....	<b>6</b>
Exemples.....	6
Ecrire dans un fichier unique.....	6
Ecrire un fichier journal par jour.....	6
<b>Chapitre 4: Extension de NLog</b> .....	<b>8</b>
Exemples.....	8
Créer une cible personnalisée.....	8
Créer un moteur de rendu de mise en page personnalisé.....	9
<b>Crédits</b> .....	<b>10</b>

# À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [nlog](#)

It is an unofficial and free nlog ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official nlog.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

# Chapitre 1: Démarrer avec nlog

## Remarques

NLog est une bibliothèque de journalisation gratuite pour .NET. Il est sous licence BSD-3, ce qui signifie qu'il peut être utilisé dans des applications commerciales.

La source de NLog peut être trouvée sur [GitHub](#)

## Versions

Versions majeures:

version	notes de version	date de sortie
4	<a href="#">notes de version</a>	2015-06-09
3	<a href="#">notes de version</a>	2014-06-02
2	<a href="#">notes de version</a>	2011-07-18
1	<a href="#">notes de version</a>	2011-01-07

## Exemples

### Écrire les premiers journaux

Installez le package [NLog.Config](#) . (Les utilisateurs de NuGet 3 doivent installer [NLog](#) et ajouter la configuration manuellement.) Ecrire dans le journal:

```
using NLog;

public class MyClass
{
    private static Logger logger = LogManager.GetCurrentClassLogger();

    public void MyMethod1()
    {
        logger.Trace("Sample trace message");
        logger.Debug("Sample debug message");
        logger.Info("Sample informational message");
        logger.Warn("Sample warning message");
        logger.Error("Sample error message");
        logger.Fatal("Sample fatal error message");

        // alternatively you can call the Log() method
        // and pass log level as the parameter.
        logger.Log(LogLevel.Info, "Sample informational message");
    }
}
```

```

try
{
    SendMail();
}
catch(Exception ex)
{
    //example writing exception
    logger.Error(ex, "Error when sending mail");
}
}
}

```

Cet exemple de configuration doit être ajouté au package NLog.Config. Sinon, ajoutez ce qui suit à la racine de l'application - ou à côté du fichier .dll en tant que "nlog.config"

```

<?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"
xsi:schemaLocation="http://www.nlog-project.org/schemas/NLog.xsd NLog.xsd"
autoReload="true"
throwExceptions="false"
internalLogLevel="Off" internalLogFile="c:\temp\nlog-internal.log">

<!-- optional, add some variables
https://github.com/nlog/NLog/wiki/Configuration-file#variables
-->
<variable name="myvar" value="myvalue"/>

<!--
See https://github.com/nlog/nlog/wiki/Configuration-file
for information on customizing logging rules and outputs.
-->
<targets>

<!--
add your targets here
See https://github.com/nlog/NLog/wiki/Targets for possible targets.
See https://github.com/nlog/NLog/wiki/Layout-Renderers for the possible layout renderers.
-->

<!-- Write events to a file with the date in the filename. -->
<target xsi:type="File" name="f" fileName="${basedir}/logs/${shortdate}.log"
layout="${longdate} ${uppercase:${level}} ${message} ${exception}" />

</targets>

<rules>
<!-- add your logging rules here -->

<!--
Write all events with minimal level of Debug (So Debug, Info, Warn, Error and Fatal, but
not Trace) to "f" -->
<logger name="*" minlevel="Debug" writeTo="f" />

</rules>
</nlog>

```

La configuration écrit les messages du journal dans un fichier.

Lire Démarrer avec nlog en ligne: <https://riptutorial.com/fr/nlog/topic/7133/demarrer-avec-nlog>

# Chapitre 2: Dépannage de NLog

## Exemples

### Activer le journal interne dans nlog.config

En cas de problème, activez l'enregistreur interne

```
<nlog internalLogFile="c:\log.txt" internalLogLevel="Trace">
  <targets>
    <!-- target configuration here -->
  </targets>
  <rules>
    <!-- log routing rules -->
  </rules>
</nlog>
```

### Activer l'enregistreur interne par programmation

En cas de problème, activez l'enregistreur interne.

C # exemple:

```
// set internal log level
InternalLogger.LogLevel = LogLevel.Trace;

// enable one of the targets: file, console, logwriter:

// enable internal logging to a file (absolute or relative path. Don't use layout renderers)
InternalLogger.LogFile = "c:\\log.txt";

// enable internal logging to the console
InternalLogger.LogToConsole = true;

// enable internal logging to the console (error stream)
InternalLogger.LogToConsoleError = true

// enable internal logging to the Trace
InternalLogger.LogToTrace = true

// enable internal logging to a custom TextWriter
InternalLogger.LogWriter = new StringWriter(); //e.g. TextWriter writer =
File.CreateText("C:\\perl.txt")
```

Lire Dépannage de NLog en ligne: <https://riptutorial.com/fr/nlog/topic/7190/depannage-de-nlog>

# Chapitre 3: Ecrire des journaux dans des fichiers

## Exemples

### Ecrire dans un fichier unique

Écrivez tous les événements avec le niveau: debug, info, warn, error et fatal dans un fichier:

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

  <targets>
    <target name="file" xsi:type="File"
            layout="${longdate} ${logger} ${message} ${exception}"
            fileName="${basedir}/logs/logfile.txt" />
  </targets>

  <rules>
    <logger name="*" minlevel="Debug" writeTo="file" />
  </rules>
</nlog>
```

### Ecrire un fichier journal par jour

Créez un fichier journal pour chaque jour. Les fichiers journaux auront les noms suivants (dépend de la culture)

```
2016-06-05.log
2016-06-06.log
2016-06-07.log
...
```

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

  <targets>
    <target name="file" xsi:type="File"
            layout="${longdate} ${logger} ${message} ${exception}"
            fileName="${basedir}/${shortdate}.log" />
  </targets>

  <rules>
    <logger name="*" minlevel="Debug" writeTo="file" />
  </rules>
</nlog>
```

Lire Ecrire des journaux dans des fichiers en ligne: <https://riptutorial.com/fr/nlog/topic/7192/ecrire->



[des-journaux-dans-des-fichiers](#)

# Chapitre 4: Extension de NLog

## Exemples

### Créer une cible personnalisée

```
using NLog;
using NLog.Config;
using NLog.Targets;

namespace MyNamespace
{
    [Target("MyFirst")]
    public sealed class MyFirstTarget: TargetWithLayout //or inherit from Target
    {
        public MyFirstTarget()
        {
            //set defaults
            this.Host = "localhost";
        }

        [RequiredParameter]
        public string Host { get; set; }

        protected override void Write(LogEventInfo logEvent)
        {
            string logMessage = this.Layout.Render(logEvent);

            //TODO write to target
        }
    }
}
```

Enregistrez-le sous le nom `MyFirst` - dès que possible - par exemple dans `main()` , `application_start()` .

```
ConfigurationItemFactory
    .Default
    .Targets
    .RegisterDefinition("MyFirst", typeof(MyNamespace.MyFirstTarget));
```

### Usage:

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

    <targets>
        <target name="target1" type="MyFirst" Host="somehost.com" />
    </targets>

    <rules>
        <logger name="*" minlevel="Debug" writeTo="target1" />
    </rules>
</nlog>
```

```
</rules>
</nlog>
```

## Créer un moteur de rendu de mise en page personnalisé

```
[LayoutRenderer("hello-world")]
public class HelloWorldLayoutRenderer : LayoutRenderer
{
    /// <summary>
    /// I'm option and not required or default
    /// </summary>
    public string Config1 { get; set; }

    /// <summary>
    /// I'm required option. And error will be thrown if not set.
    /// </summary>
    [RequiredParameter]
    public string Config2 { get; set; }

    /// <summary>
    /// I'm the default parameter.
    /// The first parameter value (without name) will be set to this one
    /// You can set me as required also.
    /// </summary>
    [DefaultParameter]
    public bool Caps {get;set;}

    protected override void Append(StringBuilder builder, LogEventInfo logEvent)
    {
        //TODO use options
        builder.Append("hello world!");
    }
}
```

Inscrivez-vous dès que possible, par exemple dans `main()` , `application_start()` .

```
//register under "hello-world"
ConfigurationItemFactory.Default.LayoutRenderers
    .RegisterDefinition("hello-world",
typeof(MyNamespace.HelloWorldLayoutRenderer ));
```

### Usage

```
 ${hello-world} - raises exception: required parameter Config2 isn't set
 ${hello-world:Config2=abc} - OK, Config2 property set
 ${hello-world:true:config2=abc} - default parameter (Caps) set to true
 ${hello-world:true:config2=abc:config1=yes} - all the three properties set.
```

Lire Extension de NLog en ligne: <https://riptutorial.com/fr/nlog/topic/7194/extension-de-nlog>

---

# Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec nlog	<a href="#">4444</a> , <a href="#">Community</a> , <a href="#">Julian</a>
2	Dépannage de NLog	<a href="#">Julian</a>
3	Ecrire des journaux dans des fichiers	<a href="#">Julian</a>
4	Extension de NLog	<a href="#">Julian</a>