

# Ngine DLL – Version 1.0.0.2

Generic functions for .Net Applications

## Contents

Introduction .....	1
Requirements.....	1
How to use the Ngine.dll.....	2
Adding the reference to your project .....	2
Basic Usage .....	3
Details of Functions.....	4
Encryption .....	4
SQL .....	4
Useful .....	6
Licence .....	6
Disclaimer.....	6

## Introduction

The Ngine.dll is a compiled unit of code built for the .Net 4 Framework and comprises of useful and generic functionality. The DLL has three namespaces (categories) of functions namely:-

Encryption	Encrypt and Decrypt strings (based on the <a href="#">RijndaelManaged</a> class)
SQL	Functionality for SQL Server connections
Useful	A collection of useful functions

## Requirements

- .Net 4
- Visual Studio 2010

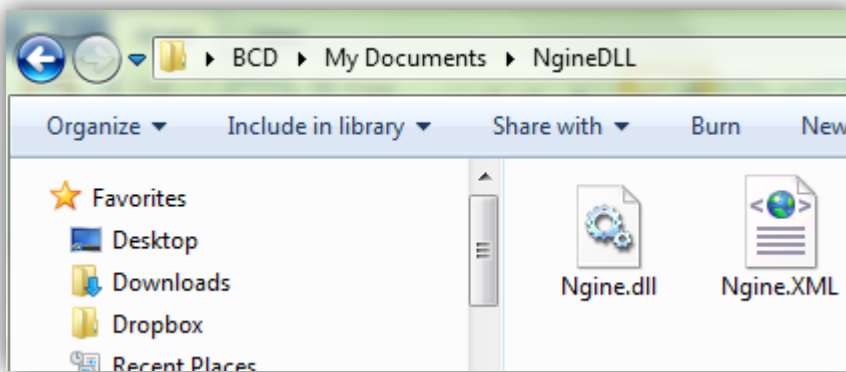
## How to use the Ngine.dll

The section below will explain how to implement Ngine.dll into your application and show some simple examples of how to use it. The examples given are using Visual Studio 2010 and show both C# and Visual Basic syntax.

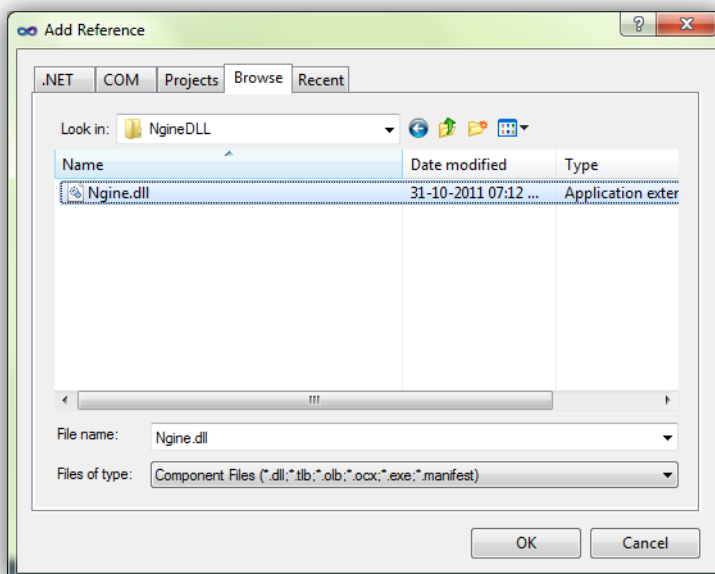
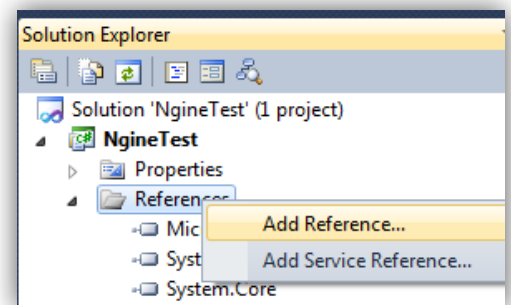
### Adding the reference to your project

Save the Ngine.dll to a location on your development machine. (Save the Ngine.XML to the same directory too for added code completion hints and Object Browser details)

For this example I have saved the files to a folder called "NgineDLL" in "My Documents"



In a Visual Studio application, right-click on the References (in your Solution Explorer) and select "Add Reference..."



Now select the dll from the location where you saved it

## Basic Usage

The Ngine is very easy to use and the examples below will show a few simple calls to the functions. Note the comments to find out more about the usage.

### C#

```
//You can either include the "using clause",
//or reference the full namespace in your code
//eg. Ngine.Useful.IsValidEmailAddress
using Ngine;

//The "Useful" namespace can be used directly without needing to
//initialise any Ngine object variable
//eg.
if (Useful.IsValidEmailAddress("bvh@f"))
{
    MessageBox.Show("Valid Email");
}
else
{
    MessageBox.Show("Not valid!");
}

//The "SQL" namespace can be used directly without needing to
//initialise any Ngine object variable
//eg.
DataSet ds = SQL.BuildDataSetFromSQL("Data Source=(local);Initial
Catalog=MyDatabase;Integrated Security=True", "SELECT * FROM MyTable");

//Encryption namespace
//To use this you need to create a new Encryption object and use it's Encrypt and
Decrypt functions
//See documentation for further info on these...
Encryption Ncrypt = new Encryption();
string Test = Ncrypt.EncryptString("Hello World");
string Test2 = Ncrypt.DecryptString(Test);
MessageBox.Show(Test2);
```

### Visual Basic

```
'You can either include the "Imports clause",
'or reference the full namespace in your code
'eg. Ngine.Useful.IsValidEmailAddress
Imports Ngine

'The "Useful" namespace can be used directly without needing to
'initialise any Ngine object variable
'eg.
If (Useful.IsValidEmailAddress("bvh@f")) Then
    MessageBox.Show("Valid Email")
Else
    MessageBox.Show("Not valid!")
End If

'The "SQL" namespace can be used directly without needing to
'initialise any Ngine object variable
'eg.
```

```
Dim ds As DataSet = SQL.BuildDataSetFromSQL("Data Source=(local);Initial
Catalog=MyDatabase;Integrated Security=True", "SELECT * FROM MyTable")

'Encryption namespace
'To use this you need to create a new Encryption object and use it's Encrypt and
Decrypt functions
'See documentation for further info on these...
Dim Ncrypt As Encryption = New Encryption()
Dim Test As String = Ncrypt.EncryptString("Hello World")
Dim Test2 As String = Ncrypt.DecryptString(Test)
MessageBox.Show(Test2)
```

## Details of Functions

### Encryption

**It is important to note that values encrypted with this function can only ever be decrypted using the Decrypt function provided in the Ngine DLL!**

As shown in the example, one must create a new instance of the Encryption object in order to use the functions.

Function	Info
<b>DecryptString</b>	Will decrypt any string that was encrypted with the function below.
<b>EncryptString</b>	Encrypt any given string using a built in "key" and "vector". Once encrypted, one can only decrypt it using the Ngine DLL. Uses .Net framework Cryptography and RijndaelManaged classes

### SQL

These functions can be used without needing to create an instance of the Ngine object.

Most of the functions require the same input parameters namely :-

ConnectionString	A valid connection string to a SQL Server
SQL	A valid SQL statement
Timeout (optional)	The SQL timeout (in seconds)

Function	Info
<b>BuildDataSetFromSQL</b>	Returns a DataSet populated with one Table containing the resultset of the given SQL statement
<b>BuildTableFromSQL</b>	Returns a Table populated with the resultset of the given SQL statement
<b>CreateConnectionString</b>	Given the basic inputs this will create a connection string to a SQL Server
<b>ExecuteSQL</b>	Executes the SQL statement and returns a Boolean value to indicate success
<b>ExecuteSQLReturnError</b>	Executes the SQL statement and returns a string message containing any error which may have occurred. Empty string

	indicates no errors
<b>ExecuteSQL (with object array)</b>	<p>Executes the SQL statement and returns a Boolean value to indicate success The SQL should be in the format eg. "UPDATE Users SET UserName=@UserName WHERE ID=@ID" The objects array should contain the values (in same order as used in SQL statement)</p> <pre>//SQL With Parameters string sql = "UPDATE Users SET UserName = @UserName, " +             "Firstname = @Firstname"; object[] obj = new object[2]; obj[0] = "user_1"; obj[1] = "Bob";  SQL.ExecuteSQL("Data Source=(local);Initial Catalog=MyDatabase;Integrated Security=True", sql, obj);</pre>
<b>ExecuteSQLReturnError (with object array)</b>	Same as above but will return the string message of any error which may have occurred
<b>ExecuteStoredProcedure</b>	<p>Executes the stored procedure and returns a Boolean value to indicate success The objects array should contain the values (in same order as used in expected by stored procedure)</p> <pre>//SQL With Parameters object[] obj = new object[2]; obj[0] = "user_1"; obj[1] = "Bob";  SQL.ExecuteSQL("Data Source=(local);Initial Catalog=MyDatabase;Integrated Security=True", "spMyProcedure", obj);</pre>
<b>ExecuteStoredProcedure ReturnError</b>	Same as above but will return the string message of any error which may have occurred
<b>ExecuteSQLScalar</b>	Executes the SQL statement and returns the first value as a string
<b>GetBoolValueFromSQL</b>	Returns the first column of the first row of the given SQL statement as a bool value (defaults to false)
<b>GetDateTimeValueFromSQL</b>	Returns the first column of the first row of the given SQL statement as a DateTime value (defaults to DateTime.MinValue)
<b>GetIntValueFromSQL</b>	Returns the first column of the first row of the given SQL statement as an int value (defaults to 0)
<b>GetLongValueFromSQL</b>	Returns the first column of the first row of the given SQL statement as a long value (defaults to 0)
<b>GetStringValueFromSQL</b>	Returns the first column of the first row of the given SQL statement as a string value

## Useful

These functions can be used without needing to create an instance of the Ngine object.

Function	Info
<b>AreByteArraysEqual</b>	Given two byte arrays, return a bool value to indicate whether or not they are equal (byte per byte)
<b>ForceToBool</b>	Attempt to convert a string value to a bool (default to false) Caters for "1" and "yes"
<b>ForceToInt</b>	Attempt to convert a string value to an int (default to 0)
<b>ForceToLong</b>	Attempt to convert a string value to a long value (default to 0)
<b>IsNumeric</b>	Given a string value will determine if it is numeric (using the .Net .ToDecimal functionality)
<b>IsValidEmailAddress</b>	Will determine whether a given string value is a valid email address
<b>RandomPassword</b>	Returns a random password of a given length. Can include UpperCase as well as numbers.

## Licence

This software may be freely distributed as part of your application installations and does not contain any restrictive licensing modules. This software may however not be copied to others who intend using it as part of their software development. This software is strictly sold "per developer" and may be used by a single user on multiple machines. Each developer is required to purchase their own licence.

## Disclaimer

THIS SOFTWARE IS PROVIDED BY NVEST DEVELOPMENT SOLUTIONS (PTY) LTD "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.