Neitzel Library

v 4.6.1.3

# Abstract

The NeitzelLib is a small library of function that I found useful for my own development. It is published under the MIT license so that it can be used freely at no cost. Please check the License.txt for the exact license agreement.

**Important Note:** I am also extending types of the .Net Framework Library. I am aware that this is not a good design for 3d party libraries but as I said already: The main intention is my own development.

Content

[Abstract 1](#_Toc449551987)

[Dependencies to other libraries 3](#_Toc449551988)

[DockPanel Suite 3](#_Toc449551989)

[Homepage and git location: 3](#_Toc449551990)

[Neitzel Assembly 4](#_Toc449551991)

[Neitzel Namespace 4](#_Toc449551992)

[DisposableObject class 4](#_Toc449551993)

[ExceptionExtensions class 4](#_Toc449551994)

[StringExtensions class 4](#_Toc449551995)

[TraceSourceExtensions class 4](#_Toc449551996)

[TypeExtensions class 4](#_Toc449551997)

[Neitzel.Network Namespace 5](#_Toc449551998)

[TcpIpClientConnection class 5](#_Toc449551999)

[TcpIpServerConnection class 5](#_Toc449552000)

[BytesReceivedEventArgs class 5](#_Toc449552001)

[NetworkConstants class 5](#_Toc449552002)

[StringDecoder class 5](#_Toc449552003)

[TextMessageEventArgs class 5](#_Toc449552004)

[Neitzel.Forms Assembly 6](#_Toc449552005)

[Neitzel.Forms Namespace 6](#_Toc449552006)

[FormsApplication 6](#_Toc449552007)

[Neitzel.Forms.Example Assembly 7](#_Toc449552008)

[FormsApplication 7](#_Toc449552009)

[Neitzel.Irc Assembly 8](#_Toc449552010)

[Neitzel.Win32 Assembly (legacy code!) 9](#_Toc449552011)

# Dependencies to other libraries

## DockPanel Suite

The DockPanel Suite was added to the Dependencies folder. The DLL was build from the source from git with target Framework 4.6.1.

This component is under MIT License. Please see WeifenLuo.WinFormsUI.Docking.license.txt for more details.

### Homepage and git location:

<http://dockpanelsuite.com/>

<https://github.com/dockpanelsuite/dockpanelsuite>

# Neitzel Assembly

## Neitzel Namespace

### DisposableObject class

The disposable object class implements IDisposable and offers the main functionality:

* Dispose() method
* Protected virtual Dispose(bool disposing) method which the deriving class should override.   
  **Do not forget to call base.Dispose(disposing)!**
* A Property Disposed which can be used to check whether the instance was disposed already.

### ExceptionExtensions class

Extends the Exception class.

Adds a GetFullMessage() method to an exception to get message and stack trace of the exception and all inner exception in one single string.

### StringExtensions class

Extends the String class.

Adds a Contains method that checks if one of multiple characters is inside the string.

Adds a FirstIndexOf method that checks the first occurance of any character given in an array of characters.

### TraceSourceExtensions class

Extends the TraceSource class.

Adds the methods Verbose, Information, Warning, Error and Critical to trace a message with less parameters and offer the option to also trace an exception together with the message.

### TypeExtensions class

Extends the Type class.

Adds a method to create an instance of a class which class name was given as parameter. This class must have a constructor that takes no arguments.

## Neitzel.Network Namespace

### TcpIpClientConnection class

A simple implementation of a TcpIp client connection which could be used to connect to a server.

Received Messages will trigger a MessageReceived event.

### TcpIpServerConnection class

A simple implementation of a TcpIp server connection. It can open a socket and wait for connections. All connections are handled on one single thread.

Received Messages will trigger a MessageReceived event with the ClientConnection (internal class) which received the message, as sender.

### BytesReceivedEventArgs class

The class for the event arguments of the MessageReceived event containing the byte array with the received bytes.

### NetworkConstants class

Some constants used inside the Neitzel.Network classes.

### StringDecoder class

An implementation of a decoder which can receive MessageReceived events and will trigger MessageDecoded events with the decoded text. It takes a newline as a separator between messages.

Right now it is unclear if this design makes any sense!

### TextMessageEventArgs class

The class for the StringDecoder.MessageDecoded event argument.

# Neitzel.Forms Assembly

## Neitzel.Forms Namespace

### FormsApplication

The FormsApplication is a small class that takes control over the ApplicationContext of the Windows Forms application.

As long as a registered form exist (is not closed / disposed), the Application will continue running.

To use it, all that must be done is

1. Inside the Program.cs / Main method, the call to Application.Run must be replaced with a call to FormsApplication.Run.
2. When additional Forms are opened, FormsApplication.AddForm(Form) must be called to register the form.

The example Neitzel.Forms.Example is showing a simple Forms application in which you can open more and more windows (forms) and it continues to run till the last window is closed.

# Neitzel.Forms.Example Assembly

This is a windows forms application which shows the extensions of the Neitzel.Forms assembly.

## FormsApplication

The example uses the FormsApplication to keep running till all forms are closed.

# Neitzel.Irc Assembly

This is the Irc Library that I started in the past and stopped working on. I just moved it in here so the code does not get lost. It is not usable right now but a few concepts could be seen that I wanted to implement.

One Unit-Test is not working because the method behind is still not implemented (NotImplementedException is thrown).  
Another Unit-Tests needs a local IRC server running to complete.

# Neitzel.Win32 Assembly (legacy code!)

This assembly contains legacy code. It can access Win32 application windows and control them. The simple parts of Win32Window might be useful if it is required to minimize / maximize / hide windows. But the code goes much deeper e.g. controlling controls inside windows.