

Namespace MatrixFFN

Classes

[FFN](#)

FFN = feed-forward-network. Ein klassisches sigmoides Netzwerk mit automatischer Datennormalisierung.

Ich werde alles mit 'decimal' realisieren - dachte ich. Dieser Wertetyp wird aber nicht von 'Math' unterstützt und ist eine Baustelle. So muß es mit 'double' gehen.

[FFN_ILGPU](#)

ILGPU-Variante der Klasse 'FFN'

FFN = feed-forward-network. Ein klassisches sigmoides Netzwerk mit automatischer Datennormalisierung.

Ich werde alles mit 'decimal' realisieren - dachte ich. Dieser Wertetyp wird aber nicht von 'Math' unterstützt und ist eine Baustelle. So muß es mit 'double' gehen.

[FFN_Window](#)

Interaktionslogik für FFN_Window.xaml

Class FFN

Namespace: [MatrixFFN](#)

Assembly: MatrixFFN.dll

FFN = feed-forward-network. Ein klassisches sigmoids Netzwerk mit automatischer Datennormalisierung.

Ich werde alles mit 'decimal' realisieren - dachte ich. Dieser Wertetyp wird aber nicht von 'Math' unterstützt und ist eine Baustelle. So muß es mit 'double' gehen.

```
public class FFN
```

Inheritance

[object](#) ← FFN

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

FFN(int[], bool, string)

Der Konstruktor der ein neues Netzwerk initialisiert.

Die Liste der Schichten muß immer mindestens 3 groß sein, da immer mindestens eine Hidden-Schicht gewollt ist.

```
public FFN(int[] layersIn, bool normalise, string name = "FFN.netz")
```

Parameters

layersIn [int](#)[]

Bsp.: schichten = { 2, 3, 1 } oder = { 3, 15, 7, 2 }

normalise [bool](#)

Neu: Datennormalisierung mittels des 'DatenNetz'es.

name [string](#)

Dateiname für die Dateioperationen

FFN(string)

Dieser Konstruktor lädt ein bereits gespeichertes Netzwerk.

```
public FFN(string name)
```

Parameters

name [string](#)

der zu nehmende Dateiname

Fields

epochesNumber

Zähler für die Lernepochen.

```
public long epochesNumber
```

Field Value

[long](#)

errorMedian

Hier ist die Fehlersumme durch alle Epochen geteilt.

```
public double errorMedian
```

Field Value

[double](#)

errorSum

Ergibt sich beim 'Fit'ten und ist die Summe über alle Epochen.

```
public double errorSum
```

Field Value

[double](#)

fileName

Dateiname für die Dateifunktionen.

```
public string fileName
```

Field Value

[string](#)

layersTopic

Bei der Initialisierung des Netzes erhalten.

Bsp.: 3 Schichten { 2, 4, 3 } oder 2 Schichten { 3, 4 }

Beachte: erster Wert = Eingabeschicht, letzter Wert = Ausgabeschicht

```
public int[] layersTopic
```

Field Value

[int](#)[]

listEpochs

Epochenliste aus der Lebenszeit im Training erhalten.

```
public List<long> listEpochs
```

Field Value

[List](#) <[long](#)>

listErrorAmount

Fehlerliste aus der Lebenszeit im Training erhalten.

```
public List<double> listErrorAmount
```

Field Value

[List](#) <[double](#)>

myData

NPOI-wrapper to read/write Excel-files

```
public NPOIexcel myData
```

Field Value

NPOIexcel

stopWatchFit

Zeitstopper für Fit

```
public Stopwatch stopWatchFit
```

Field Value

[StopWatch](#)

stopWatchPredict

Zeitstopper für Predict

```
public Stopwatch stopWatchPredict
```

Field Value

[StopWatch](#)

stopWatchTrain

Zeitstopper für Train

```
public Stopwatch stopWatchTrain
```

Field Value

[StopWatch](#)

timeFit

Zeitausgabe von stopWatchFit

```
public string timeFit
```

Field Value

[string](#) ↗

timePredict

Zeitausgabe von stopWatchPredict

```
public string timePredict
```

Field Value

[string ↗](#)

timeTrain

Zeitausgabe von stopWatchTrain

```
public string timeTrain
```

Field Value

[string ↗](#)

workingTopic

String für das Fontend für die Topic.

```
public string workingTopic
```

Field Value

[string ↗](#)

Methods

DataNetInit(double[][], double[][])

Hier werden automatisch aus den übergebenen Daten die beiden DatenNetze gebildet.

Bisher werden keine kategorischen Daten verarbeitet. (ToDo-Liste)

```
public void DataNetInit(double[][] dataInputArray, double[][] dataOutputArray)
```

Parameters

dataInputArray [double](#)[][]

die Eingaben ins Netz als Feld

dataOutputArray [double](#)[][]

die Ausgaben des Netzes als Feld

DeNormInputArray(double[])

Liefert über das DatenNetz (Eingang) die konvertierten Daten.

```
public double[] DeNormInputArray(double[] normedData)
```

Parameters

normedData [double](#)[]

Eingabefeld

Returns

[double](#)[]

Feld mit den entnormalisierten Werten

DeNormOutputArray(double[])

Liefert über das DatenNetz (Ausgang) die konvertierten Daten.

```
public double[] DeNormOutputArray(double[] normedData)
```

Parameters

`normedData` [double](#)[]

Eingabefeld

Returns

[double](#)[]

Feld mit den entnormalisierten Werten

Fit(double[][], double[][], long)

Trainiert ganze Epochen (komplette Datensätze) und präsentiert alle Daten pro Epoche einmal zufällig.

Normalisiert die Daten automatisch (empfohlen)

```
public string Fit(double[][] inputArrayField, double[][] outputArrayField, long epochsIn)
```

Parameters

`inputArrayField` [double](#)[][]

Eingabedaten

`outputArrayField` [double](#)[][]

Ausgabedaten

`epochsIn` [long](#)

Anzahl der Epochen

Returns

[string](#)

result of the training

Fit_LocalData(long)

Trainiert ganze Epochen (komplette Datensätze) und präsentiert alle Daten pro Epoche einmal zufällig.

Uses the dataset loaded with 'LoadDataFromExcel()'

Normalisiert die Daten automatisch (empfohlen)

```
public string Fit_LocalData(long epochsIn)
```

Parameters

epochsIn [long](#)

Anzahl der Epochen

Returns

[string](#)

result of the training

InitLocalData()

For convenience and speed the local data can now be prepared. 'Fit_LocalData' and 'Train_LocalData' can benefit from it.

```
public void InitLocalData()
```

LoadData(string)

Lädt ein gespeichertes Netzwerk. Dies kann als Konstruktorauftrag laufen bzw. auch mittendrin. Das gesamte Netzwerk wird anhand der gespeicherten Daten neu initialisiert - kann seine Form also komplett ändern.

```
public bool LoadData(string name)
```

Parameters

name [string](#)

Dateiname

Returns

[bool](#)

Funktionserfolg

LoadDataFromExcel(string, int, bool, int, int)

Loads the training data from an Excel-file. Always asks for the filename with dialog.

```
public void LoadDataFromExcel(string fileName, int sheetNumber = 0, bool useHeaders = true,  
int noOfInputs = 0, int noOfOutputs = 0)
```

Parameters

[fileName](#) [string](#)

if given silent mode will be used

[sheetNumber](#) [int](#)

the sheets number in the workbook

[useHeaders](#) [bool](#)

using a header line of cells?

[noOfInputs](#) [int](#)

optional you give it manually

[noOfOutputs](#) [int](#)

optional you give it manually

NormInputArray(double[])

Liefert über das DatenNetz (Eingang) die konvertierten Daten.

```
public double[] NormInputArray(double[] realData)
```

Parameters

`realData double[]`

Eingabefeld

Returns

`double[]`

Feld mit den normalisierten Werten

NormOutputArray(double[])

Liefert über das DatenNetz (Ausgang) die konvertierten Daten.

`public double[] NormOutputArray(double[] realData)`

Parameters

`realData double[]`

Eingabefeld

Returns

`double[]`

Feld mit den normalisierten Werten

ParseTopic()

Hilfsfunktion für das Fontend, damit nach dem Laden die Oberfläche auch alle Daten bekommt.

Aus der Netzttopic (int[]) wird der Darstellungsstring erstellt.

`public void ParseTopic()`

PartArray(double[], int, int, ref double[], ref double[])

Splits the datafield into two parts. Needed for the splitting of the input data read from file (Excel).

Best used with PartsNums().

```
public static void PartArray(double[] fullData, int firstSize, int secondSize, ref double[] firstPart, ref double[] secondPart)
```

Parameters

fullData [double\[\]](#)

read data'block

firstSize [int](#)

counted inputs from headers

secondSize [int](#)

counted outputs from headers

firstPart [double\[\]](#)

split field beginning

secondPart [double\[\]](#)

split field continuing

PartsNums(string[], ref int, ref int)

Counts the headers in-/outputs.

```
public static void PartsNums(string[] headers, ref int inputs, ref int outputs)
```

Parameters

headers [string\[\]](#)

the loaded headers from the excel file

inputs int[]

number of them

outputs int[]

number of them

Predict(double[])

Diese Funktion führt einen 'feed forward predict' mit den übergebenen Daten aus.

```
public double[] Predict(double[] inputLayer)
```

Parameters

inputLayer double[]

Eingangsdaten in das Netz

Returns

double[]

Ausgangsdaten des Netzes

SaveData(string)

Speichert das Netzwerk (alle Variablen). Beachte: mindestens einmal muß 'Train' absolviert werden, damit überhaupt etwas zum Speichern existiert.

```
public bool SaveData(string name)
```

Parameters

name string[]

erwünschter Dateiname

Returns

bool ↗

Funktionserfolg

SetLearningRate(double)

Multipliziert die Lernwerte mit dem übergebenen Wert.

Nützlich für die Adaption der Lernschritte zum Bsp. mittels des Fehlerwertes. Skaliert die bisherigen Alphawerte.

Falls schon einmal verändert, wird die alte Korrektur rückgängig gemacht und der neue Wert benutzt.

```
public void SetLearningRate(double target)
```

Parameters

target double ↗

Train(double[], double[])

Trainiert einmal das Netzwerk mit dem übergebenen Ein-/Ausgabemuster.

Gibt den 'mean square sum'-Fehler zurück.

```
public double Train(double[] inputLayer, double[] outputLayer)
```

Parameters

inputLayer double []

Netzwerkeingang

outputLayer double []

Netzwerkausgang

Returns

double ↗

Liefert den Fehlerwert zurück.

Train_LocalData(int)

Trainiert einmal das Netzwerk mit dem übergebenen Ein-/Ausgabemuster.

Gibt den 'mean square sum'-Fehler zurück.

```
public double Train_LocalData(int sampleN)
```

Parameters

sampleN int ↗

number of the local datarow

Returns

double ↗

Liefert den Fehlerwert zurück.

Class FFN_ILGPU

Namespace: [MatrixFFN](#)

Assembly: MatrixFFN.dll

ILGPU-Variante der Klasse 'FFN'

FFN = feed-forward-network. Ein klassisches sigmoids Netzwerk mit automatischer Datennormalisierung.

Ich werde alles mit 'decimal' realisieren - dachte ich. Dieser Wertetyp wird aber nicht von 'Math' unterstützt und ist eine Baustelle. So muß es mit 'double' gehen.

```
public class FFN_ILGPU
```

Inheritance

[object](#) ← FFN_ILGPU

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

FFN_ILGPU(int[], bool, string)

Der Konstruktor der ein neues Netzwerk initialisiert.

Die Liste der Schichten muß immer mindestens 3 groß sein, da immer mindestens eine Hidden-Schicht gewollt ist.

```
public FFN_ILGPU(int[] layersIn, bool normalise, string name = "FFN_ILGPU.netz")
```

Parameters

layersIn [int](#)[]

Bsp.: schichten = { 2, 3, 1 } oder = { 3, 15, 7, 2 }

normalise [bool](#)

Neu: Datennormalisierung mittels des 'DatenNetz'es.

name [string](#)

Dateiname für die Dateioperationen

FFN_ILGPU(string)

Dieser Konstruktor lädt ein bereits gespeichertes Netzwerk.

```
public FFN_ILGPU(string name)
```

Parameters

name [string](#)

der zu nehmende Dateiname

Fields

epochesNumber

Zähler für die Lernepochen.

```
public long epochesNumber
```

Field Value

[long](#)

errorMedian

Hier ist die Fehlersumme durch alle Epochen geteilt.

```
public double errorMedian
```

Field Value

[double](#)

errorSum

Ergibt sich beim 'Fit'ten und ist die Summe über alle Epochen.

```
public double errorSum
```

Field Value

[double](#)

fileName

Dateiname für die Dateifunktionen.

```
public string fileName
```

Field Value

[string](#)

layersTopic

Bei der Initialisierung des Netzes erhalten.

Bsp.: 3 Schichten { 2, 4, 3 } oder 2 Schichten { 3, 4 }

Beachte: erster Wert = Eingabeschicht, letzter Wert = Ausgabeschicht

```
public int[] layersTopic
```

Field Value

[int](#)[]

listEpochs

Epochenliste aus der Lebenszeit im Training erhalten.

```
public List<long> listEpochs
```

Field Value

[List](#) <[long](#)>

listErrorAmount

Fehlerliste aus der Lebenszeit im Training erhalten.

```
public List<double> listErrorAmount
```

Field Value

[List](#) <[double](#)>

myData

NPOI-wrapper to read/write Excel-files

```
public NPOIexcel myData
```

Field Value

NPOIexcel

stopWatchFit

Zeitstopper für Fit

```
public Stopwatch stopWatchFit
```

Field Value

[StopWatch](#)

stopWatchPredict

Zeitstopper für Predict

```
public Stopwatch stopWatchPredict
```

Field Value

[StopWatch](#)

stopWatchTrain

Zeitstopper für Train

```
public Stopwatch stopWatchTrain
```

Field Value

[StopWatch](#)

timeFit

Zeitausgabe von stopWatchFit

```
public string timeFit
```

Field Value

[string](#) ↗

timePredict

Zeitausgabe von stopWatchPredict

```
public string timePredict
```

Field Value

[string ↗](#)

timeTrain

Zeitausgabe von stopWatchTrain

```
public string timeTrain
```

Field Value

[string ↗](#)

workingTopic

String für das Fontend für die Topic.

```
public string workingTopic
```

Field Value

[string ↗](#)

Methods

DataNetInit(double[][], double[][])

Hier werden automatisch aus den übergebenen Daten die beiden DatenNetze gebildet.

Bisher werden keine kategorischen Daten verarbeitet. (ToDo-Liste)

```
public void DataNetInit(double[][] dataInputArray, double[][] dataOutputArray)
```

Parameters

dataInputArray double[][]

die Eingaben ins Netz als Feld

dataOutputArray double[][]

die Ausgaben des Netzes als Feld

DeNormInputArray(double[])

Liefert über das DatenNetz (Eingang) die konvertierten Daten.

```
public double[] DeNormInputArray(double[] normedData)
```

Parameters

normedData double[][]

Eingabefeld

Returns

double[][]

Feld mit den entnormalisierten Werten

DeNormOutputArray(double[])

Liefert über das DatenNetz (Ausgang) die konvertierten Daten.

```
public double[] DeNormOutputArray(double[] normedData)
```

Parameters

`normedData` [double^{\[\]}](#)

Eingabefeld

Returns

[double^{\[\]}](#)

Feld mit den entnormalisierten Werten

Fit(double[][][], double[][][], long)

Trainiert ganze Epochen (komplette Datensätze) und präsentiert alle Daten pro Epoche einmal zufällig.

Normalisiert die Daten automatisch (empfohlen)

```
public string Fit(double[][][] inputArrayField, double[][][] outputArrayField, long epochsIn)
```

Parameters

`inputArrayField` [double^{\[\]\[\]}](#)

Eingabedaten

`outputArrayField` [double^{\[\]\[\]}](#)

Ausgabedaten

`epochsIn` [long^{\[\]}](#)

Anzahl der Epochen

Returns

[string^{\[\]}](#)

Fit_LocalData(long)

Trainiert ganze Epochen (komplette Datensätze) und präsentiert alle Daten pro Epoche einmal zufällig.

Uses the dataset loaded with 'LoadDataFromExcel()'

Normalisiert die Daten automatisch (empfohlen)

```
public string Fit_LocalData(long epochsIn)
```

Parameters

epochsIn [long](#)

Anzahl der Epochen

Returns

[string](#)

result of the training

InitLocalData()

For convenience and speed the local data can now be prepared. 'Fit_LocalData' and 'Train_LocalData' can benefit from it.

```
public void InitLocalData()
```

LoadData(string)

Lädt ein gespeichertes Netzwerk. Dies kann als Konstruktorauftrag laufen bzw. auch mittendrin. Das gesamte Netzwerk wird anhand der gespeicherten Daten neu initialisiert - kann seine Form also komplett ändern.

```
public bool LoadData(string name)
```

Parameters

name [string](#)

Dateiname

Returns

[bool](#)

Funktionserfolg

LoadDataFromExcel(string, int, bool, int, int)

Loads the training data from an Excel-file. Always asks for the filename with dialog.

```
public void LoadDataFromExcel(string fileName, int sheetNumber = 0, bool useHeaders = true,  
int noOfInputs = 0, int noOfOutputs = 0)
```

Parameters

[fileName](#) [string](#)

if given silent mode will be used

[sheetNumber](#) [int](#)

the sheets number in the workbook

[useHeaders](#) [bool](#)

using a header line of cells?

[noOfInputs](#) [int](#)

optional you give it manually

[noOfOutputs](#) [int](#)

optional you give it manually

NormInputArray(double[])

Liefert über das DatenNetz (Eingang) die konvertierten Daten.

```
public double[] NormInputArray(double[] realData)
```

Parameters

`realData double[]`

Eingabefeld

Returns

`double[]`

Feld mit den normalisierten Werten

NormOutputArray(double[])

Liefert über das DatenNetz (Ausgang) die konvertierten Daten.

`public double[] NormOutputArray(double[] realData)`

Parameters

`realData double[]`

Eingabefeld

Returns

`double[]`

Feld mit den normalisierten Werten

ParseTopic()

Hilfsfunktion für das Fontend, damit nach dem Laden die Oberfläche auch alle Daten bekommt.

Aus der Netzttopic (int[]) wird der Darstellungsstring erstellt.

`public void ParseTopic()`

PartArray(double[], int, int, ref double[], ref double[])

Splits the datafield into two parts. Needed for the splitting of the input data read from file (Excel).

Best used with PartsNums().

```
public static void PartArray(double[] fullData, int firstSize, int secondSize, ref double[] firstPart, ref double[] secondPart)
```

Parameters

fullData [double](#)[]

read data`block

firstSize [int](#)

counted inputs from headers

secondSize [int](#)

counted outputs from headers

firstPart [double](#)[]

split field beginning

secondPart [double](#)[]

split field continuing

PartsNums(string[], ref int, ref int)

Counts the headers in-/outputs.

```
public static void PartsNums(string[] headers, ref int inputs, ref int outputs)
```

Parameters

headers [string](#)[]

the loaded headers from the excel file

inputs [int](#)

number of them

outputs [int](#)

number of them

Predict(double[])

Diese Funktion führt einen 'feed forward predict' mit den übergebenen Daten aus.

```
public double[] Predict(double[] inputLayer)
```

Parameters

[inputLayer](#) [double](#)[]

Eingangsdaten in das Netz

Returns

[double](#)[]

Ausgangsdaten des Netzes

SaveData(string)

Speichert das Netzwerk (alle Variablen). Beachte: mindestens einmal muß 'Train' absolviert werden, damit überhaupt etwas zum Speichern existiert.

```
public bool SaveData(string name)
```

Parameters

[name](#) [string](#)

erwünschter Dateiname

Returns

bool ↗

Funktionserfolg

SetLearningRate(double)

Multipliziert die Lernwerte mit dem übergebenen Wert.

Nützlich für die Adaption der Lernschritte zum Bsp. mittels des Fehlerwertes. Skaliert die bisherigen Alphawerte.

Falls schon einmal verändert, wird die alte Korrektur rückgängig gemacht und der neue Wert benutzt.

```
public void SetLearningRate(double target)
```

Parameters

target double ↗

Train(double[], double[])

Trainiert einmal das Netzwerk mit dem übergebenen Ein-/Ausgabemuster.

Gibt den 'mean square sum'-Fehler zurück.

```
public double Train(double[] inputLayer, double[] outputLayer)
```

Parameters

inputLayer double ↗ []

Netzwerkeingang

outputLayer double ↗ []

Netzwerkausgang

Returns

double ↗

Liefert den Fehlerwert zurück.

Train_LocalData(int)

Trainiert einmal das Netzwerk mit dem übergebenen Ein-/Ausgabemuster.

Gibt den 'mean square sum'-Fehler zurück.

```
public double Train_LocalData(int sampleN)
```

Parameters

sampleN [int](#)

number of the local datarow

Returns

[double](#)

Liefert den Fehlerwert zurück.

Class FFN_Window

Namespace: [MatrixFFN](#)

Assembly: MatrixFFN.dll

Interaktionslogik für FFN_Window.xaml

```
public class FFN_Window : Window, IAnimatable, ISupportInitialize, IFrameworkInputElement,  
IInputElement, IQueryAmbient, IAddChild, IComponentConnector
```

Inheritance

[object](#) ← [DispatcherObject](#) ← [DependencyObject](#) ← [Visual](#) ← [UIElement](#) ←
[FrameworkElement](#) ← [Control](#) ← [ContentControl](#) ← [Window](#) ← FFN_Window

Implements

[IAnimatable](#), [ISupportInitialize](#), [IFrameworkInputElement](#), [IInputElement](#), [IQueryAmbient](#),
[IAddChild](#), [IComponentConnector](#)

Inherited Members

[Window.AllowsTransparencyProperty](#), [Window.DpiChangedEvent](#), [Window.IconProperty](#),
[Window.IsActiveProperty](#), [Window.LeftProperty](#), [Window.ResizeModeProperty](#),
[Window>ShowActivatedProperty](#), [Window>ShowInTaskbarProperty](#),
[Window.SizeToContentProperty](#), [Window.TaskBarItemInfoProperty](#), [Window.TitleProperty](#),
[Window.TopmostProperty](#), [Window.TopProperty](#), [Window.WindowStateProperty](#),
[Window.WindowStyleProperty](#), [Window.Activate\(\)](#), [Window.ArrangeOverride\(Size\)](#),
[Window.Close\(\)](#), [Window.DragMove\(\)](#), [Window.GetWindow\(DependencyObject\)](#),
[Window.Hide\(\)](#), [Window.MeasureOverride\(Size\)](#), [Window.OnActivated\(EventArgs\)](#),
[Window.OnClosed\(EventArgs\)](#), [Window.OnClosing\(CancelEventArgs\)](#),
[Window.OnContentChanged\(object, object\)](#), [Window.OnContentRendered\(EventArgs\)](#),
[Window.OnCreateAutomationPeer\(\)](#), [Window.OnDeactivated\(EventArgs\)](#),
[Window.OnDpiChanged\(DpiScale, DpiScale\)](#), [Window.OnLocationChanged\(EventArgs\)](#),
[Window.OnManipulationBoundaryFeedback\(ManipulationBoundaryFeedbackEventArgs\)](#),
[Window.OnSourceInitialized\(EventArgs\)](#), [Window.OnStateChanged\(EventArgs\)](#),
[Window.OnVisualChildrenChanged\(DependencyObject, DependencyObject\)](#),
[Window.OnVisualParentChanged\(DependencyObject\)](#), [Window.Show\(\)](#), [Window.ShowDialog\(\)](#),
[WindowAllowsTransparency](#), [Window.DialogResult](#), [Window.Icon](#), [Window.IsActive](#),
[Window.Left](#), [Window.LogicalChildren](#), [Window.OwnedWindows](#), [Window.Owner](#),
[Window.ResizeMode](#), [Window.RestoreBounds](#), [Window>ShowActivated](#),
[Window>ShowInTaskbar](#), [Window.SizeToContent](#), [Window.TaskBarItemInfo](#), [Window.Title](#),
[Window.Top](#), [Window.Topmost](#), [Window.WindowStartupLocation](#), [Window.WindowState](#),

[Window.WindowStyle](#) , [Window.Activated](#) , [Window.Closed](#) , [Window.Closing](#) ,
[Window.ContentRendered](#) , [Window.Deactivated](#) , [Window.DpiChanged](#) ,
[Window.LocationChanged](#) , [Window.SourceInitialized](#) , [Window.StateChanged](#) ,
[ContentControl.ContentProperty](#) , [ContentControl.ContentStringFormatProperty](#) ,
[ContentControl.ContentTemplateProperty](#) , [ContentControl.ContentTemplateSelectorProperty](#) ,
[ContentControl.HasContentProperty](#) , [ContentControl.AddChild\(object\)](#) ,
[ContentControl.AddText\(string\)](#) , [ContentControl.OnContentStringFormatChanged\(string, string\)](#) ,
[ContentControl.OnContentTemplateChanged\(DataTemplate, DataTemplate\)](#) ,
[ContentControl.OnContentTemplateSelectorChanged\(DataTemplateSelector, DataTemplateSelector\)](#) ,
[ContentControl.Content](#) , [ContentControl.ContentStringFormat](#) , [ContentControl.ContentTemplate](#) ,
[ContentControl.ContentTemplateSelector](#) , [ContentControl.HasContent](#) ,
[Control.BackgroundProperty](#) , [Control.BorderBrushProperty](#) , [Control.BorderThicknessProperty](#) ,
[Control.FontFamilyProperty](#) , [Control.FontSizeProperty](#) , [Control.FontStretchProperty](#) ,
[Control.FontStyleProperty](#) , [Control.FontWeightProperty](#) , [Control.ForegroundProperty](#) ,
[Control.HorizontalContentAlignmentProperty](#) , [Control.IsTabStopProperty](#) ,
[Control.MouseDoubleClickEvent](#) , [Control.PaddingProperty](#) ,
[Control.PreviewMouseDoubleClickEvent](#) , [Control.TabIndexProperty](#) , [Control.TemplateProperty](#) ,
[Control.VerticalContentAlignmentProperty](#) , [Control.OnMouseDoubleClick\(MouseEventArgs\)](#) ,
[Control.OnPreviewMouseDoubleClick\(MouseEventArgs\)](#) ,
[Control.OnTemplateChanged\(ControlTemplate, ControlTemplate\)](#) , [Control.ToString\(\)](#) ,
[Control.Background](#) , [Control.BorderBrush](#) , [Control.BorderThickness](#) , [Control.FontFamily](#) ,
[Control.FontSize](#) , [Control.FontStretch](#) , [Control.FontStyle](#) , [Control.FontWeight](#) ,
[Control.Foreground](#) , [Control.HandlesScrolling](#) , [Control.HorizontalContentAlignment](#) ,
[Control.IsTabStop](#) , [Control.Padding](#) , [Control.TabIndex](#) , [Control.Template](#) ,
[Control.VerticalContentAlignment](#) , [Control.MouseDoubleClick](#) ,
[Control.PreviewMouseDoubleClick](#) , [FrameworkElement.ActualHeightProperty](#) ,
[FrameworkElement.ActualWidthProperty](#) , [FrameworkElement.BindingGroupProperty](#) ,
[FrameworkElement.ContextMenuClosingEvent](#) , [FrameworkElement.ContextMenuOpeningEvent](#) ,
[FrameworkElement.ContextMenuProperty](#) , [FrameworkElement.CursorProperty](#) ,
[FrameworkElement.DataContextProperty](#) , [FrameworkElement.DefaultStyleKeyProperty](#) ,
[FrameworkElement.FlowDirectionProperty](#) , [FrameworkElement.FocusVisualStyleProperty](#) ,
[FrameworkElement.ForceCursorProperty](#) , [FrameworkElement.HeightProperty](#) ,
[FrameworkElement.HorizontalAlignmentProperty](#) , [FrameworkElement.InputScopeProperty](#) ,
[FrameworkElement.LanguageProperty](#) , [FrameworkElement.LayoutTransformProperty](#) ,
[FrameworkElement.LoadedEvent](#) , [FrameworkElement.MarginProperty](#) ,
[FrameworkElement.MaxHeightProperty](#) , [FrameworkElement.MaxWidthProperty](#) ,
[FrameworkElement.MinHeightProperty](#) , [FrameworkElement.MinWidthProperty](#) ,
[FrameworkElement.NameProperty](#) , [FrameworkElement.OverridesDefaultStyleProperty](#) ,
[FrameworkElement.RequestBringIntoViewEvent](#) , [FrameworkElement.SizeChangedEvent](#) ,
[FrameworkElement.StyleProperty](#) , [FrameworkElement.TagProperty](#) ,

[FrameworkElement.ToolTipClosingEvent](#) , [FrameworkElement.ToolTipOpeningEvent](#) ,
[FrameworkElement.ToolTipProperty](#) , [FrameworkElement.UnloadedEvent](#) ,
[FrameworkElement.UseLayoutRoundingProperty](#) , [FrameworkElement.VerticalAlignmentProperty](#) ,
[FrameworkElement.WidthProperty](#) , [FrameworkElement.AddLogicalChild\(object\)](#) ,
[FrameworkElement.ApplyTemplate\(\)](#) , [FrameworkElement.ArrangeCore\(Rect\)](#) ,
[FrameworkElement.BeginInit\(\)](#) , [FrameworkElement.BeginStoryboard\(Storyboard\)](#) ,
[FrameworkElement.BeginStoryboard\(Storyboard, HandoffBehavior\)](#) ,
[FrameworkElement.BeginStoryboard\(Storyboard, HandoffBehavior, bool\)](#) ,
[FrameworkElement.BringIntoView\(\)](#) , [FrameworkElement.BringIntoView\(Rect\)](#) ,
[FrameworkElement.EndInit\(\)](#) , [FrameworkElement.FindName\(string\)](#) ,
[FrameworkElement.FindResource\(object\)](#) ,
[FrameworkElement.GetBindingExpression\(DependencyProperty\)](#) ,
[FrameworkElement.GetFlowDirection\(DependencyObject\)](#) , [FrameworkElement.GetLayoutClip\(Size\)](#) ,
[FrameworkElement.GetTemplateChild\(string\)](#) , [FrameworkElement.GetUIParentCore\(\)](#) ,
[FrameworkElement.GetVisualChild\(int\)](#) , [FrameworkElement.MeasureCore\(Size\)](#) ,
[FrameworkElement.MoveFocus\(TraversalRequest\)](#) , [FrameworkElement.OnApplyTemplate\(\)](#) ,
[FrameworkElement.OnContextMenuClosing\(ContextMenuEventArgs\)](#) ,
[FrameworkElement.OnContextMenuOpening\(ContextMenuEventArgs\)](#) ,
[FrameworkElement.OnGotFocus\(RoutedEventArgs\)](#) , [FrameworkElement.OnInitialized\(EventArgs\)](#) ,
[FrameworkElement.OnPropertyChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[FrameworkElement.OnRenderSizeChanged\(SizeChangedEventArgs\)](#) ,
[FrameworkElement.OnStyleChanged\(Style, Style\)](#) ,
[FrameworkElement.OnToolTipClosing\(ToolTipEventArgs\)](#) ,
[FrameworkElement.OnToolTipOpening\(ToolTipEventArgs\)](#) ,
[FrameworkElement.ParentLayoutInvalidated\(UIElement\)](#) ,
[FrameworkElement.PredictFocus\(FocusNavigationDirection\)](#) ,
[FrameworkElement.RegisterName\(string, object\)](#) , [FrameworkElement.RemoveLogicalChild\(object\)](#) ,
[FrameworkElement.SetBinding\(DependencyProperty, string\)](#) ,
[FrameworkElement.SetBinding\(DependencyProperty, BindingBase\)](#) ,
[FrameworkElement.SetFlowDirection\(DependencyObject, FlowDirection\)](#) ,
[FrameworkElement.SetResourceReference\(DependencyProperty, object\)](#) ,
[FrameworkElement.TryFindResource\(object\)](#) , [FrameworkElement.UnregisterName\(string\)](#) ,
[FrameworkElement.UpdateDefaultStyle\(\)](#) , [FrameworkElement.ActualHeight](#) ,
[FrameworkElement.ActualWidth](#) , [FrameworkElement.BindingGroup](#) ,
[FrameworkElement.ContextMenu](#) , [FrameworkElement.Cursor](#) , [FrameworkElement.DataContext](#) ,
[FrameworkElement.DefaultStyleKey](#) , [FrameworkElement.FlowDirection](#) ,
[FrameworkElement.FocusVisualStyle](#) , [FrameworkElement.ForceCursor](#) , [FrameworkElement.Height](#) ,
[FrameworkElement.HorizontalContentAlignment](#) , [FrameworkElement.InheritanceBehavior](#) ,
[FrameworkElement.InputScope](#) , [FrameworkElement.IsInitialized](#) , [FrameworkElement.IsLoaded](#) ,
[FrameworkElement.Language](#) , [FrameworkElement.LayoutTransform](#) , [FrameworkElement.Margin](#) ,

[FrameworkElement.MaxHeight](#) , [FrameworkElement.MaxWidth](#) , [FrameworkElement.MinHeight](#) ,
[FrameworkElement.MinWidth](#) , [FrameworkElement.Name](#) ,
[FrameworkElementOverridesDefaultStyle](#) , [FrameworkElement.Parent](#) ,
[FrameworkElement.Resources](#) , [FrameworkElement.Style](#) , [FrameworkElement.Tag](#) ,
[FrameworkElement.TemplatedParent](#) , [FrameworkElement.ToolTip](#) , [FrameworkElement.Triggers](#) ,
[FrameworkElement.UseLayoutRounding](#) , [FrameworkElement.VerticalAlignment](#) ,
[FrameworkElement.VisualChildrenCount](#) , [FrameworkElement.Width](#) ,
[FrameworkElement.ContextMenuClosing](#) , [FrameworkElement.ContextMenuOpening](#) ,
[FrameworkElement.DataContextChanged](#) , [FrameworkElement.Initialized](#) ,
[FrameworkElement.Loaded](#) , [FrameworkElement.RequestBringIntoView](#) ,
[FrameworkElement.SizeChanged](#) , [FrameworkElement.SourceUpdated](#) ,
[FrameworkElement.TargetUpdated](#) , [FrameworkElement.ToolTipClosing](#) ,
[FrameworkElement.ToolTipOpening](#) , [FrameworkElement.Unloaded](#) ,
[UIElement.AllowDropProperty](#) , [UIElement.AreAnyTouchesCapturedProperty](#) ,
[UIElement.AreAnyTouchesCapturedWithinProperty](#) , [UIElement.AreAnyTouchesDirectlyOverProperty](#) ,
[UIElement.AreAnyTouchesOverProperty](#) , [UIElement.BitmapEffectInputProperty](#) ,
[UIElement.BitmapEffectProperty](#) , [UIElement.CacheModeProperty](#) , [UIElement.ClipProperty](#) ,
[UIElement.ClipToBoundsProperty](#) , [UIElement.DragEnterEvent](#) , [UIElement.DragLeaveEvent](#) ,
[UIElement.DragOverEvent](#) , [UIElement.DropEvent](#) , [UIElement.EffectProperty](#) ,
[UIElement.FocusableProperty](#) , [UIElement.GiveFeedbackEvent](#) , [UIElement.GotFocusEvent](#) ,
[UIElement.GotKeyboardFocusEvent](#) , [UIElement.GotMouseCaptureEvent](#) ,
[UIElement.GotStylusCaptureEvent](#) , [UIElement.GotTouchCaptureEvent](#) ,
[UIElement.IsEnabledProperty](#) , [UIElement.IsFocusedProperty](#) , [UIElement.IsHitTestVisibleProperty](#) ,
[UIElement.IsKeyboardFocusedProperty](#) , [UIElement.IsKeyboardFocusWithinProperty](#) ,
[UIElement.IsManipulationEnabledProperty](#) , [UIElement.IsMouseCapturedProperty](#) ,
[UIElement.IsMouseCaptureWithinProperty](#) , [UIElement.IsMouseDirectlyOverProperty](#) ,
[UIElement.IsMouseOverProperty](#) , [UIElement.IsStylusCapturedProperty](#) ,
[UIElement.IsStylusCaptureWithinProperty](#) , [UIElement.IsStylusDirectlyOverProperty](#) ,
[UIElement.IsStylusOverProperty](#) , [UIElement.IsVisibleProperty](#) , [UIElement.KeyDownEvent](#) ,
[UIElement.KeyUpEvent](#) , [UIElement.LostFocusEvent](#) , [UIElement.LostKeyboardFocusEvent](#) ,
[UIElement.LostMouseCaptureEvent](#) , [UIElement.LostStylusCaptureEvent](#) ,
[UIElement.LostTouchCaptureEvent](#) , [UIElement.ManipulationBoundaryFeedbackEvent](#) ,
[UIElement.ManipulationCompletedEvent](#) , [UIElement.ManipulationDeltaEvent](#) ,
[UIElement.ManipulationInertiaStartingEvent](#) , [UIElement.ManipulationStartedEvent](#) ,
[UIElement.ManipulationStartingEvent](#) , [UIElement.MouseDownEvent](#) , [UIElement.MouseEnterEvent](#) ,
[UIElement.MouseLeaveEvent](#) , [UIElement.MouseLeftButtonDownEvent](#) ,
[UIElement.MouseLeftButtonUpEvent](#) , [UIElement.MouseMoveEvent](#) ,
[UIElement.MouseRightButtonDownEvent](#) , [UIElement.MouseRightButtonUpEvent](#) ,
[UIElement.MouseUpEvent](#) , [UIElement.MouseWheelEvent](#) , [UIElement.OpacityMaskProperty](#) ,
[UIElement.OpacityProperty](#) , [UIElement.PreviewDragEnterEvent](#) ,

[UIElement.PreviewDragLeaveEvent](#) , [UIElement.PreviewDragOverEvent](#) ,
[UIElement.PreviewDropEvent](#) , [UIElement.PreviewGiveFeedbackEvent](#) ,
[UIElement.PreviewGotKeyboardFocusEvent](#) , [UIElement.PreviewKeyDownEvent](#) ,
[UIElement.PreviewKeyUpEvent](#) , [UIElement.PreviewLostKeyboardFocusEvent](#) ,
[UIElement.PreviewMouseDownEvent](#) , [UIElement.PreviewMouseLeftButtonDownEvent](#) ,
[UIElement.PreviewMouseLeftButtonUpEvent](#) , [UIElement.PreviewMouseMoveEvent](#) ,
[UIElement.PreviewMouseRightButtonDownEvent](#) , [UIElement.PreviewMouseRightButtonUpEvent](#) ,
[UIElement.PreviewMouseUpEvent](#) , [UIElement.PreviewMouseWheelEvent](#) ,
[UIElement.PreviewQueryContinueDragEvent](#) , [UIElement.PreviewStylusButtonDownEvent](#) ,
[UIElement.PreviewStylusButtonUpEvent](#) , [UIElement.PreviewStylusDownEvent](#) ,
[UIElement.PreviewStylusInAirMoveEvent](#) , [UIElement.PreviewStylusInRangeEvent](#) ,
[UIElement.PreviewStylusMoveEvent](#) , [UIElement.PreviewStylusOutOfRangeEvent](#) ,
[UIElement.PreviewStylusSystemGestureEvent](#) , [UIElement.PreviewStylusUpEvent](#) ,
[UIElement.PreviewTextInputEvent](#) , [UIElement.PreviewTouchDownEvent](#) ,
[UIElement.PreviewTouchMoveEvent](#) , [UIElement.PreviewTouchUpEvent](#) ,
[UIElement.QueryContinueDragEvent](#) , [UIElement.QueryCursorEvent](#) ,
[UIElement.RenderTransformOriginProperty](#) , [UIElement.RenderTransformProperty](#) ,
[UIElement.SnapsToDevicePixelsProperty](#) , [UIElement.StylusButtonDownEvent](#) ,
[UIElement.StylusButtonUpEvent](#) , [UIElement.StylusDownEvent](#) , [UIElement.StylusEnterEvent](#) ,
[UIElement.StylusInAirMoveEvent](#) , [UIElement.StylusInRangeEvent](#) , [UIElement.StylusLeaveEvent](#) ,
[UIElement.StylusMoveEvent](#) , [UIElement.StylusOutOfRangeEvent](#) ,
[UIElement.StylusSystemGestureEvent](#) , [UIElement.StylusUpEvent](#) , [UIElement.TextInputEvent](#) ,
[UIElement.TouchDownEvent](#) , [UIElement.TouchEnterEvent](#) , [UIElement.TouchLeaveEvent](#) ,
[UIElement.TouchMoveEvent](#) , [UIElement.TouchUpEvent](#) , [UIElement.UidProperty](#) ,
[UIElement.VisibilityProperty](#) , [UIElement.AddHandler\(RoutedEvent, Delegate\)](#) ,
[UIElement.AddHandler\(RoutedEvent, Delegate, bool\)](#) ,
[UIElement.AddToEventRoute\(EventRoute, RoutedEventArgs\)](#) ,
[UIElement.ApplyAnimationClock\(DependencyProperty, AnimationClock\)](#) ,
[UIElement.ApplyAnimationClock\(DependencyProperty, AnimationClock, HandoffBehavior\)](#) ,
[UIElement.Arrange\(Rect\)](#) , [UIElement.BeginAnimation\(DependencyProperty, AnimationTimeline\)](#) ,
[UIElement.BeginAnimation\(DependencyProperty, AnimationTimeline, HandoffBehavior\)](#) ,
[UIElement.CaptureMouse\(\)](#) , [UIElement.CaptureStylus\(\)](#) , [UIElement.CaptureTouch\(TouchDevice\)](#) ,
[UIElement.Focus\(\)](#) , [UIElement.GetAnimationBaseValue\(DependencyProperty\)](#) ,
[UIElement.HitTestCore\(GeometryHitTestParameters\)](#) , [UIElement.HitTestCore\(PointHitTestParameters\)](#) ,
[UIElement.InputHitTest\(Point\)](#) , [UIElement.InvalidateArrange\(\)](#) , [UIElement.InvalidateMeasure\(\)](#) ,
[UIElement.InvalidateVisual\(\)](#) , [UIElement.Measure\(Size\)](#) ,
[UIElement.OnAccessKey\(AccessKeyEventArgs\)](#) , [UIElement.OnChildDesiredSizeChanged\(UIElement\)](#) ,
[UIElement.OnDragEnter\(DragEventArgs\)](#) , [UIElement.OnDragLeave\(DragEventArgs\)](#) ,
[UIElement.OnDragOver\(DragEventArgs\)](#) , [UIElement.OnDrop\(DragEventArgs\)](#) ,
[UIElement.OnGiveFeedback\(GiveFeedbackEventArgs\)](#) ,

[UIElement.OnGotKeyboardFocus\(KeyboardFocusChangedEventArgs\)](#) ,
[UIElement.OnGotMouseCapture\(MouseEventArgs\)](#) ,
[UIElement.OnGotStylusCapture\(StylusEventArgs\)](#) , [UIElement.OnGotTouchCapture\(TouchEventArgs\)](#) ,
[UIElement.OnIsKeyboardFocusedChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsKeyboardFocusWithinChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsMouseCapturedChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsMouseCaptureWithinChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsMouseDirectlyOverChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsStylusCapturedChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsStylusCaptureWithinChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsStylusDirectlyOverChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnKeyDown\(KeyEventArgs\)](#) , [UIElement.OnKeyUp\(KeyEventArgs\)](#) ,
[UIElement.OnLostFocus\(RoutedEventArgs\)](#) ,
[UIElement.OnLostKeyboardFocus\(KeyboardFocusChangedEventArgs\)](#) ,
[UIElement.OnLostMouseCapture\(MouseEventArgs\)](#) ,
[UIElement.OnLostStylusCapture\(StylusEventArgs\)](#) , [UIElement.OnLostTouchCapture\(TouchEventArgs\)](#) ,
[UIElement.OnManipulationCompleted\(ManipulationCompletedEventArgs\)](#) ,
[UIElement.OnManipulationDelta\(ManipulationDeltaEventArgs\)](#) ,
[UIElement.OnManipulationInertiaStarting\(ManipulationInertiaStartingEventArgs\)](#) ,
[UIElement.OnManipulationStarted\(ManipulationStartedEventArgs\)](#) ,
[UIElement.OnManipulationStarting\(ManipulationStartingEventArgs\)](#) ,
[UIElement.OnMouseDown\(MouseButtonEventArgs\)](#) , [UIElement.OnMouseEnter\(MouseEventArgs\)](#) ,
[UIElement.OnMouseLeave\(MouseEventArgs\)](#) ,
[UIElement.OnMouseLeftButtonDown\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseLeftButtonUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseMove\(MouseEventArgs\)](#) ,
[UIElement.OnMouseRightButtonDown\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseRightButtonUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseWheel\(MouseEventArgs\)](#) ,
[UIElement.OnPreviewDragEnter\(DragEventArgs\)](#) , [UIElement.OnPreviewDragLeave\(DragEventArgs\)](#) ,
[UIElement.OnPreviewDragOver\(DragEventArgs\)](#) , [UIElement.OnPreviewDrop\(DragEventArgs\)](#) ,
[UIElement.OnPreviewGiveFeedback\(GiveFeedbackEventArgs\)](#) ,
[UIElement.OnPreviewGotKeyboardFocus\(KeyboardFocusChangedEventArgs\)](#) ,
[UIElement.OnPreviewKeyDown\(KeyEventArgs\)](#) , [UIElement.OnPreviewKeyUp\(KeyEventArgs\)](#) ,
[UIElement.OnPreviewLostKeyboardFocus\(KeyboardFocusChangedEventArgs\)](#) ,
[UIElement.OnPreviewMouseDown\(MouseButtonEventArgs\)](#) ,
[UIElement.OnPreviewMouseLeftButtonDown\(MouseButtonEventArgs\)](#) ,
[UIElement.OnPreviewMouseLeftButtonUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnPreviewMouseMove\(MouseEventArgs\)](#) ,

[UIElement.OnPreviewMouseRightButtonDown\(MouseButtonEventArgs\)](#) ,
[UIElement.OnPreviewMouseRightButtonUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnPreviewMouseUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnPreviewMouseWheel\(MouseWheelEventArgs\)](#) ,
[UIElement.OnPreviewQueryContinueDrag\(QueryContinueDragEventArgs\)](#) ,
[UIElement.OnPreviewStylusButtonDown\(StylusButtonEventArgs\)](#) ,
[UIElement.OnPreviewStylusButtonUp\(StylusButtonEventArgs\)](#) ,
[UIElement.OnPreviewStylusDown\(StylusDownEventArgs\)](#) ,
[UIElement.OnPreviewStylusInAirMove\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewStylusInRange\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewStylusMove\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewStylusOutOfRange\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewStylusSystemGesture\(StylusSystemGestureEventArgs\)](#) ,
[UIElement.OnPreviewStylusUp\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewTextInput\(TextCompositionEventArgs\)](#) ,
[UIElement.OnPreviewTouchDown\(TouchEventArgs\)](#) ,
[UIElement.OnPreviewTouchMove\(TouchEventArgs\)](#) , [UIElement.OnPreviewTouchUp\(TouchEventArgs\)](#) ,
[UIElement.OnQueryContinueDrag\(QueryContinueDragEventArgs\)](#) ,
[UIElement.OnQueryCursor\(QueryCursorEventArgs\)](#) , [UIElement.OnRender\(DrawingContext\)](#) ,
[UIElement.OnStylusButtonDown\(StylusButtonEventArgs\)](#) ,
[UIElement.OnStylusButtonUp\(StylusButtonEventArgs\)](#) ,
[UIElement.OnStylusDown\(StylusDownEventArgs\)](#) , [UIElement.OnStylusEnter\(StylusEventArgs\)](#) ,
[UIElement.OnStylusInAirMove\(StylusEventArgs\)](#) , [UIElement.OnStylusInRange\(StylusEventArgs\)](#) ,
[UIElement.OnStylusLeave\(StylusEventArgs\)](#) , [UIElement.OnStylusMove\(StylusEventArgs\)](#) ,
[UIElement.OnStylusOutOfRange\(StylusEventArgs\)](#) ,
[UIElement.OnStylusSystemGesture\(StylusSystemGestureEventArgs\)](#) ,
[UIElement.OnStylusUp\(StylusEventArgs\)](#) , [UIElement.OnTextInput\(TextCompositionEventArgs\)](#) ,
[UIElement.OnTouchDown\(TouchEventArgs\)](#) , [UIElement.OnTouchEnter\(TouchEventArgs\)](#) ,
[UIElement.OnTouchLeave\(TouchEventArgs\)](#) , [UIElement.OnTouchMove\(TouchEventArgs\)](#) ,
[UIElement.OnTouchUp\(TouchEventArgs\)](#) , [UIElement.RaiseEvent\(RoutedEventArgs\)](#) ,
[UIElement.ReleaseAllTouchCaptures\(\)](#) , [UIElement.ReleaseMouseCapture\(\)](#) ,
[UIElement.ReleaseStylusCapture\(\)](#) , [UIElement.ReleaseTouchCapture\(TouchDevice\)](#) ,
[UIElement.RemoveHandler\(RoutedEvent, Delegate\)](#) , [UIElement.TranslatePoint\(Point, UIElement\)](#) ,
[UIElement.UpdateLayout\(\)](#) , [UIElement.AllowDrop](#) , [UIElement.AreAnyTouchesCaptured](#) ,
[UIElement.AreAnyTouchesCapturedWithin](#) , [UIElement.AreAnyTouchesDirectlyOver](#) ,
[UIElement.AreAnyTouchesOver](#) , [UIElement.BitmapEffect](#) , [UIElement.BitmapEffectInput](#) ,
[UIElement.CacheMode](#) , [UIElement.Clip](#) , [UIElement.ClipToBounds](#) ,
[UIElement.CommandBindings](#) , [UIElement.DesiredSize](#) , [UIElement.Effect](#) , [UIElement.Focusable](#) ,
[UIElement.HasAnimatedProperties](#) , [UIElement.HasEffectiveKeyboardFocus](#) ,
[UIElement.InputBindings](#) , [UIElement.IsArrangeValid](#) , [UIElement.IsEnabled](#) ,

[UIElement.IsEnabledCore](#) , [UIElement.IsFocused](#) , [UIElement.IsHitTestVisible](#) ,
[UIElement.IsInputMethodEnabled](#) , [UIElement.IsKeyboardFocused](#) ,
[UIElement.IsKeyboardFocusWithin](#) , [UIElement.IsManipulationEnabled](#) , [UIElement.IsMeasureValid](#) ,
[UIElement.IsMouseCaptured](#) , [UIElement.IsMouseCaptureWithin](#) , [UIElement.IsMouseDirectlyOver](#) ,
[UIElement.IsMouseOver](#) , [UIElement.IsStylusCaptured](#) , [UIElement.IsStylusCaptureWithin](#) ,
[UIElement.IsStylusDirectlyOver](#) , [UIElement.IsStylusOver](#) , [UIElement.IsVisible](#) , [UIElement.Opacity](#) ,
[UIElement.OpacityMask](#) , [UIElement.PersistId](#) , [UIElement.RenderSize](#) ,
[UIElement.RenderTransform](#) , [UIElement.RenderTransformOrigin](#) , [UIElement.SnapsToDevicePixels](#) ,
[UIElement.StylusPlugIns](#) , [UIElement.TouchesCaptured](#) , [UIElement.TouchesCapturedWithin](#) ,
[UIElement.TouchesDirectlyOver](#) , [UIElement.TouchesOver](#) , [UIElement.Uid](#) , [UIElement.Visibility](#) ,
[UIElement.DragEnter](#) , [UIElement.DragLeave](#) , [UIElement.DragOver](#) , [UIElement.Drop](#) ,
[UIElement.FocusableChanged](#) , [UIElement.GiveFeedback](#) , [UIElement.GotFocus](#) ,
[UIElement.GotKeyboardFocus](#) , [UIElement.GotMouseCapture](#) , [UIElement.GotStylusCapture](#) ,
[UIElement.GotTouchCapture](#) , [UIElement.IsEnabledChanged](#) , [UIElement.IsHitTestVisibleChanged](#) ,
[UIElement.IsKeyboardFocusedChanged](#) , [UIElement.IsKeyboardFocusWithinChanged](#) ,
[UIElement.IsMouseCapturedChanged](#) , [UIElement.IsMouseCaptureWithinChanged](#) ,
[UIElement.IsMouseDirectlyOverChanged](#) , [UIElement.IsStylusCapturedChanged](#) ,
[UIElement.IsStylusCaptureWithinChanged](#) , [UIElement.IsStylusDirectlyOverChanged](#) ,
[UIElement.IsVisibleChanged](#) , [UIElement.KeyDown](#) , [UIElement.KeyUp](#) ,
[UIElement.LayoutUpdated](#) , [UIElement.LostFocus](#) , [UIElement.LostKeyboardFocus](#) ,
[UIElement.LostMouseCapture](#) , [UIElement.LostStylusCapture](#) , [UIElement.LostTouchCapture](#) ,
[UIElement.ManipulationBoundaryFeedback](#) , [UIElement.ManipulationCompleted](#) ,
[UIElement.ManipulationDelta](#) , [UIElement.ManipulationInertiaStarting](#) ,
[UIElement.ManipulationStarted](#) , [UIElement.ManipulationStarting](#) , [UIElement.MouseDown](#) ,
[UIElement.MouseEnter](#) , [UIElement.MouseLeave](#) , [UIElement.MouseLeftButtonDown](#) ,
[UIElement.MouseLeftButtonUp](#) , [UIElement.MouseMove](#) , [UIElement.MouseRightButtonDown](#) ,
[UIElement.MouseRightButtonUp](#) , [UIElement.MouseUp](#) , [UIElement.MouseWheel](#) ,
[UIElement.PreviewDragEnter](#) , [UIElement.PreviewDragLeave](#) , [UIElement.PreviewDragOver](#) ,
[UIElement.PreviewDrop](#) , [UIElement.PreviewGiveFeedback](#) , [UIElement.PreviewGotKeyboardFocus](#) ,
[UIElement.PreviewKeyDown](#) , [UIElement.PreviewKeyUp](#) , [UIElement.PreviewLostKeyboardFocus](#) ,
[UIElement.PreviewMouseDown](#) , [UIElement.PreviewMouseLeftButtonDown](#) ,
[UIElement.PreviewMouseLeftButtonUp](#) , [UIElement.PreviewMouseMove](#) ,
[UIElement.PreviewMouseRightButtonDown](#) , [UIElement.PreviewMouseRightButtonUp](#) ,
[UIElement.PreviewMouseUp](#) , [UIElement.PreviewMouseWheel](#) ,
[UIElement.PreviewQueryContinueDrag](#) , [UIElement.PreviewStylusButtonDown](#) ,
[UIElement.PreviewStylusButtonUp](#) , [UIElement.PreviewStylusDown](#) ,
[UIElement.PreviewStylusInAirMove](#) , [UIElement.PreviewStylusInRange](#) ,
[UIElement.PreviewStylusMove](#) , [UIElement.PreviewStylusOutOfRange](#) ,
[UIElement.PreviewStylusSystemGesture](#) , [UIElement.PreviewStylusUp](#) , [UIElement.PreviewTextInput](#) ,
[UIElement.PreviewTouchDown](#) , [UIElement.PreviewTouchMove](#) , [UIElement.PreviewTouchUp](#) ,

[UIElement.QueryContinueDrag](#) , [UIElement.QueryCursor](#) , [UIElement.StylusButtonDown](#) ,
[UIElement.StylusButtonUp](#) , [UIElement.StylusDown](#) , [UIElement.StylusEnter](#) ,
[UIElement.StylusInAirMove](#) , [UIElement.StylusInRange](#) , [UIElement.StylusLeave](#) ,
[UIElement.StylusMove](#) , [UIElement.StylusOutOfRange](#) , [UIElement.StylusSystemGesture](#) ,
[UIElement.StylusUp](#) , [UIElement.TextInput](#) , [UIElement.TouchDown](#) , [UIElement.TouchEnter](#) ,
[UIElement.TouchLeave](#) , [UIElement.TouchMove](#) , [UIElement.TouchUp](#) ,
[Visual.AddVisualChild\(Visual\)](#) , [Visual.FindCommonVisualAncestor\(DependencyObject\)](#) ,
[Visual.IsAncestorOf\(DependencyObject\)](#) , [Visual.IsDescendantOf\(DependencyObject\)](#) ,
[Visual.PointFromScreen\(Point\)](#) , [Visual.PointToScreen\(Point\)](#) , [Visual.RemoveVisualChild\(Visual\)](#) ,
[Visual.TransformToAncestor\(Visual3D\)](#) , [Visual.TransformToAncestor\(Visual\)](#) ,
[Visual.TransformToDescendant\(Visual\)](#) , [Visual.TransformToVisual\(Visual\)](#) , [Visual.VisualBitmapEffect](#) ,
[Visual.VisualBitmapEffectInput](#) , [Visual.VisualBitmapScalingMode](#) , [Visual.VisualCacheMode](#) ,
[Visual.VisualClearTypeHint](#) , [Visual.VisualClip](#) , [Visual.VisualEdgeMode](#) , [Visual.VisualEffect](#) ,
[Visual.VisualOffset](#) , [Visual.VisualOpacity](#) , [Visual.VisualOpacityMask](#) , [Visual.VisualParent](#) ,
[Visual.VisualScrollableAreaClip](#) , [Visual.VisualTextHintingMode](#) , [Visual.VisualTextRenderingMode](#) ,
[Visual.VisualTransform](#) , [Visual.VisualXSnappingGuidelines](#) , [Visual.VisualYSnappingGuidelines](#) ,
[DependencyObject.ClearValue\(DependencyProperty\)](#) ,
[DependencyObject.ClearValue\(DependencyPropertyKey\)](#) ,
[DependencyObject.CoerceValue\(DependencyProperty\)](#) , [DependencyObject.Equals\(object\)](#) ,
[DependencyObject.GetHashCode\(\)](#) , [DependencyObject.GetLocalValueEnumerator\(\)](#) ,
[DependencyObject.GetValue\(DependencyProperty\)](#) ,
[DependencyObject.InvalidateProperty\(DependencyProperty\)](#) ,
[DependencyObject.ReadLocalValue\(DependencyProperty\)](#) ,
[DependencyObject.SetCurrentValue\(DependencyProperty, object\)](#) ,
[DependencyObject.SetValue\(DependencyProperty, object\)](#) ,
[DependencyObject.SetValue\(DependencyPropertyKey, object\)](#) ,
[DependencyObject.ShouldSerializeProperty\(DependencyProperty\)](#) ,
[DependencyObject.DependencyObjectType](#) , [DependencyObject.IsSealed](#) ,
[DispatcherObject.Dispatcher](#) , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

FFN_Window()

Der Konstruktor, der alle Komponenten initialisiert.

```
public FFN_Window()
```

Fields

inputArrayField

Input data for the network.

```
public double[][][] inputArrayField
```

Field Value

[double](#)[][][]

outputArrayField

Output data for the network.

```
public double[][][] outputArrayField
```

Field Value

[double](#)[][][]

Methods

GetStatusDatasetCheck()

```
public int GetStatusDatasetCheck()
```

Returns

[int](#)

InitializeComponent()

InitializeComponent

```
public void InitializeComponent()
```

SetStatusDone(string)

bewirkt Grün für den Betrachter.

```
public void SetStatusDone(string text)
```

Parameters

text [string](#)

Ausgabetext in der Leiste

SetStatusWorking(string, int)

In Arbeit soll die Farbe anders sein.

```
public void SetStatusWorking(string text, int prozent)
```

Parameters

text [string](#)

Ausgabetext in der Leiste

prozent [int](#)

prozentualer Fortschritt

Namespace MatrixFFN.Tools

Classes

[CanvasChart](#)

Using LINQ to show up to two lines in the chart in properly scaled way.

Allows to choose not to connect the datapoints with lines for unordered datasets and you can reduce the shown dataset to the last elements.

You can open a bigger window with the charts data. ('CanvasWindow')

Chart is painted onto a given 'Canvas'.

[CanvasTopic](#)

Shows the nets structure graphical in the given 'Canvas'..

Additionally you can open the 'CanvasWindow' to see a bigger 'Canvas' with the diagramm.

[CanvasWindow](#)

Interaktionslogik für CanvasFenster.xaml

[DataNet](#)

Die Klasse 'DatenNetz' normalisiert die realen Daten und zurück.

Das geht für reale (Pattern) Features.

[DataNetILGPU](#)

Die Klasse 'DatenNetz' normalisiert die realen Daten und zurück.

Das geht für reale (Pattern) Features.

[Matrix](#)

Diese Klasse bildet Matrixrechenarten für das neuronale Netz ab.

Wichtig ist hierbei die Matrixmultiplikation von 2 Matrixen, die auch völlig korrekt nach Formelsammlung erfolgt!

Unterteilung der Funktionen in:

- Operation auf die interne Matrix
- static Matrix Funktion: Operation auf Rückgabematrix
- static bool Funktion: Operation auf Zielmatrix (beste Geschwindigkeit)

MatrixILGPU

Diese Klasse bildet Matrixrechenarten für das neuronale Netz ab. Variante für ILGPU.

Wichtig ist hierbei die Matrixmultiplikation von 2 Matrixen, die auch völlig korrekt nach Formelsammlung erfolgt!

ILGPU braucht zusätzlich zur Logik die GPU-Seite, bei der über Aktionen die jeweiligen Kernel ausgeführt werden.

Unterteilung der Funktionen in:

- Operation auf die interne Matrix
- static Matrix Funktion: Operation auf Rückgabematrix
- static void Funktion: Operation auf Zielmatrix (beste Geschwindigkeit)

Pattern

Diese Klasse soll Daten für die sigmoide Funktion normalisieren. Wertebereich in den abgebildet wird ist [0.25, 0.75] - so wird eine Fehlerkorrektur auch in den Extremen ermöglicht.

Diese Klasse kann für alle Daten genommen werden. Kategorisches sollte mit Pattern_Long verarbeitet werden.

StopWatch

Ein kleine Hilfsklasse, die die Ausgabe der 'Stopwatch' schön formatiert.

Structs

ColorNumber

helper for the colordnumbers used

Class CanvasChart

Namespace: [MatrixFFN.Tools](#)

Assembly: MatrixFFN.dll

Using LINQ to show up to two lines in the chart in properly scaled way.

Allows to choose not to connect the datapoints with lines for unordered datasets and you can reduce the shown dataset to the last elements.

You can open a bigger window with the charts data. ('CanvasWindow')

Chart is painted onto a given 'Canvas'.

```
public class CanvasChart : CanvasWindow, IAnimatable, ISupportInitialize,
IFrameworkInputElement, IInputElement, IQueryAmbient, IAddChild, IComponentConnector
```

Inheritance

```
object ↗ ← DispatcherObject ↗ ← DependencyObject ↗ ← Visual ↗ ← UIElement ↗ ←
FrameworkElement ↗ ← Control ↗ ← ContentControl ↗ ← Window ↗ ← CanvasWindow ← CanvasChart
```

Implements

```
IAnimatable ↗ , ISupportInitialize ↗ , IFrameworkInputElement ↗ , IInputElement ↗ , IQueryAmbient ↗ ,
IAddChild ↗ , IComponentConnector ↗
```

Inherited Members

```
CanvasWindow.isNowToEnd , CanvasWindow.InitializeComponent() ,
WindowAllowsTransparencyProperty ↗ , Window.DpiChangedEvent ↗ , Window.IconProperty ↗ ,
Window.IsActiveProperty ↗ , Window.LeftProperty ↗ , Window.ResizeModeProperty ↗ ,
Window>ShowActivatedProperty ↗ , Window>ShowInTaskbarProperty ↗ ,
Window.SizeToContentProperty ↗ , Window.TaskBarItemInfoProperty ↗ , Window.TitleProperty ↗ ,
Window.TopmostProperty ↗ , Window.TopProperty ↗ , Window.WindowStateProperty ↗ ,
Window.WindowStyleProperty ↗ , Window.Activate() ↗ , Window.ArrangeOverride(Size) ↗ ,
Window.Close() ↗ , Window.DragMove() ↗ , Window.GetWindow(DependencyObject) ↗ ,
Window.Hide() ↗ , Window.MeasureOverride(Size) ↗ , Window.OnActivated(EventArgs) ↗ ,
Window.OnClosed(EventArgs) ↗ , Window.OnClosing(CancelEventArgs) ↗ ,
Window.OnContentChanged(object, object) ↗ , Window.OnContentRendered(EventArgs) ↗ ,
Window.OnCreateAutomationPeer() ↗ , Window.OnDeactivated(EventArgs) ↗ ,
Window.OnDpiChanged(DpiScale, DpiScale) ↗ , Window.OnLocationChanged(EventArgs) ↗ ,
Window.OnManipulationBoundaryFeedback(ManipulationBoundaryFeedbackEventArgs) ↗ ,
Window.OnSourceInitialized(EventArgs) ↗ , Window.OnStateChanged(EventArgs) ↗ ,
```

[Window.OnVisualChildrenChanged\(DependencyObject, DependencyObject\)](#) ,
[Window.OnVisualParentChanged\(DependencyObject\)](#) , [Window.Show\(\)](#) , [Window.ShowDialog\(\)](#) ,
[WindowAllowsTransparency](#) , [Window.DialogResult](#) , [Window.Icon](#) , [Window.IsActive](#) ,
[Window.Left](#) , [WindowLogicalChildren](#) , [WindowOwnedWindows](#) , [WindowOwner](#) ,
[Window.ResizeMode](#) , [Window.RestoreBounds](#) , [Window>ShowActivated](#) ,
[Window>ShowInTaskbar](#) , [Window.SizeToContent](#) , [Window.TaskBarItemInfo](#) , [Window.Title](#) ,
[Window.Top](#) , [Window.Topmost](#) , [Window.WindowStartupLocation](#) , [Window.WindowState](#) ,
[Window.WindowStyle](#) , [Window.Activated](#) , [Window.Closed](#) , [Window.Closing](#) ,
[Window.ContentRendered](#) , [Window.Deactivated](#) , [Window.DpiChanged](#) ,
[Window.LocationChanged](#) , [Window.SourceInitialized](#) , [Window.StateChanged](#) ,
[ContentControl.ContentProperty](#) , [ContentControl.ContentStringFormatProperty](#) ,
[ContentControl.ContentTemplateProperty](#) , [ContentControl.ContentTemplateSelectorProperty](#) ,
[ContentControl.HasContentProperty](#) , [ContentControl.AddChild\(object\)](#) ,
[ContentControl.AddText\(string\)](#) , [ContentControl.OnContentStringFormatChanged\(string, string\)](#) ,
[ContentControl.OnContentTemplateChanged\(DataTemplate, DataTemplate\)](#) ,
[ContentControl.OnContentTemplateSelectorChanged\(DataTemplateSelector, DataTemplateSelector\)](#) ,
[ContentControl.Content](#) , [ContentControl.ContentStringFormat](#) , [ContentControl.ContentTemplate](#) ,
[ContentControl.ContentTemplateSelector](#) , [ContentControl.HasContent](#) ,
[Control.BackgroundProperty](#) , [Control.BorderBrushProperty](#) , [Control.BorderThicknessProperty](#) ,
[Control.FontFamilyProperty](#) , [Control.FontSizeProperty](#) , [Control.FontStretchProperty](#) ,
[Control.FontStyleProperty](#) , [Control.FontWeightProperty](#) , [Control.ForegroundProperty](#) ,
[Control.HorizontalContentAlignmentProperty](#) , [Control.IsTabStopProperty](#) ,
[Control.MouseDoubleClickEvent](#) , [Control.PaddingProperty](#) ,
[Control.PreviewMouseDoubleClickEvent](#) , [Control.TabIndexProperty](#) , [Control.TemplateProperty](#) ,
[Control.VerticalContentAlignmentProperty](#) , [Control.OnMouseDoubleClick\(MouseEventArgs\)](#) ,
[Control.OnPreviewMouseDoubleClick\(MouseEventArgs\)](#) ,
[Control.OnTemplateChanged\(ControlTemplate, ControlTemplate\)](#) , [Control.ToString\(\)](#) ,
[Control.Background](#) , [Control.BorderBrush](#) , [Control.BorderThickness](#) , [Control.FontFamily](#) ,
[Control.FontSize](#) , [Control.FontStretch](#) , [Control.FontStyle](#) , [Control.FontWeight](#) ,
[Control.Foreground](#) , [Control.HandlesScrolling](#) , [Control.HorizontalContentAlignment](#) ,
[Control.IsTabStop](#) , [Control.Padding](#) , [Control.TabIndex](#) , [Control.Template](#) ,
[Control.VerticalContentAlignment](#) , [Control.MouseDoubleClick](#) ,
[Control.PreviewMouseDoubleClick](#) , [FrameworkElement.ActualHeightProperty](#) ,
[FrameworkElement.ActualWidthProperty](#) , [FrameworkElement.BindingGroupProperty](#) ,
[FrameworkElement.ContextMenuClosingEvent](#) , [FrameworkElement.ContextMenuOpeningEvent](#) ,
[FrameworkElement.ContextMenuProperty](#) , [FrameworkElement.CursorProperty](#) ,
[FrameworkElement.DataContextProperty](#) , [FrameworkElement.DefaultStyleKeyProperty](#) ,
[FrameworkElement.FlowDirectionProperty](#) , [FrameworkElement.FocusVisualStyleProperty](#) ,
[FrameworkElement.ForceCursorProperty](#) , [FrameworkElement.HeightProperty](#) ,
[FrameworkElement.HorizontalAlignmentProperty](#) , [FrameworkElement.InputScopeProperty](#) ,

[FrameworkElement.LanguageProperty](#) , [FrameworkElement.LayoutTransformProperty](#) ,
[FrameworkElement.LoadedEvent](#) , [FrameworkElement.MarginProperty](#) ,
[FrameworkElement.MaxHeightProperty](#) , [FrameworkElement.MaxWidthProperty](#) ,
[FrameworkElement.MinHeightProperty](#) , [FrameworkElement.MinWidthProperty](#) ,
[FrameworkElement.NameProperty](#) , [FrameworkElement.OverridesDefaultStyleProperty](#) ,
[FrameworkElement.RequestBringIntoViewEvent](#) , [FrameworkElement.SizeChangedEvent](#) ,
[FrameworkElement.StyleProperty](#) , [FrameworkElement.TagProperty](#) ,
[FrameworkElement.ToolTipClosingEvent](#) , [FrameworkElement.ToolTipOpeningEvent](#) ,
[FrameworkElement.ToolTipProperty](#) , [FrameworkElement.UnloadedEvent](#) ,
[FrameworkElement.UseLayoutRoundingProperty](#) , [FrameworkElement.VerticalAlignmentProperty](#) ,
[FrameworkElement.WidthProperty](#) , [FrameworkElement.AddLogicalChild\(object\)](#) ,
[FrameworkElement.ApplyTemplate\(\)](#) , [FrameworkElement.ArrangeCore\(Rect\)](#) ,
[FrameworkElement.BeginInit\(\)](#) , [FrameworkElement.BeginStoryboard\(Storyboard\)](#) ,
[FrameworkElement.BeginStoryboard\(Storyboard, HandoffBehavior\)](#) ,
[FrameworkElement.BeginStoryboard\(Storyboard, HandoffBehavior, bool\)](#) ,
[FrameworkElement.BringIntoView\(\)](#) , [FrameworkElement.BringIntoView\(Rect\)](#) ,
[FrameworkElement.EndInit\(\)](#) , [FrameworkElement.FindName\(string\)](#) ,
[FrameworkElement.FindResource\(object\)](#) ,
[FrameworkElement.GetBindingExpression\(DependencyProperty\)](#) ,
[FrameworkElement.GetFlowDirection\(DependencyObject\)](#) , [FrameworkElement.GetLayoutClip\(Size\)](#) ,
[FrameworkElement.GetTemplateChild\(string\)](#) , [FrameworkElement.GetUIParentCore\(\)](#) ,
[FrameworkElement.GetVisualChild\(int\)](#) , [FrameworkElement.MeasureCore\(Size\)](#) ,
[FrameworkElement.MoveFocus\(TraversalRequest\)](#) , [FrameworkElement.OnApplyTemplate\(\)](#) ,
[FrameworkElement.OnContextMenuClosing\(ContextMenuEventArgs\)](#) ,
[FrameworkElement.OnContextMenuOpening\(ContextMenuEventArgs\)](#) ,
[FrameworkElement.OnGotFocus\(RoutedEventArgs\)](#) , [FrameworkElement.OnInitialized\(EventArgs\)](#) ,
[FrameworkElement.OnPropertyChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[FrameworkElement.OnRenderSizeChanged\(SizeChangedEventArgs\)](#) ,
[FrameworkElement.OnStyleChanged\(Style, Style\)](#) ,
[FrameworkElement.OnToolTipClosing\(ToolTipEventArgs\)](#) ,
[FrameworkElement.OnToolTipOpening\(ToolTipEventArgs\)](#) ,
[FrameworkElement.ParentLayoutInvalidated\(UIElement\)](#) ,
[FrameworkElement.PredictFocus\(FocusNavigationDirection\)](#) ,
[FrameworkElement.RegisterName\(string, object\)](#) , [FrameworkElement.RemoveLogicalChild\(object\)](#) ,
[FrameworkElement.SetBinding\(DependencyProperty, string\)](#) ,
[FrameworkElement.SetBinding\(DependencyProperty, BindingBase\)](#) ,
[FrameworkElement.SetFlowDirection\(DependencyObject, FlowDirection\)](#) ,
[FrameworkElement.SetResourceReference\(DependencyProperty, object\)](#) ,
[FrameworkElement.TryFindResource\(object\)](#) , [FrameworkElement.UnregisterName\(string\)](#) ,
[FrameworkElement.UpdateDefaultStyle\(\)](#) , [FrameworkElement.ActualHeight](#) ,

[FrameworkElement.ActualWidth](#) , [FrameworkElement.BindingGroup](#) ,
[FrameworkElement.ContextMenu](#) , [FrameworkElement.Cursor](#) , [FrameworkElement.DataContext](#) ,
[FrameworkElement.DefaultStyleKey](#) , [FrameworkElement.FlowDirection](#) ,
[FrameworkElement.FocusVisualStyle](#) , [FrameworkElement.ForceCursor](#) , [FrameworkElement.Height](#) ,
[FrameworkElement.HorizontalAlignment](#) , [FrameworkElement.InheritanceBehavior](#) ,
[FrameworkElement.InputScope](#) , [FrameworkElement.IsInitialized](#) , [FrameworkElement.IsLoaded](#) ,
[FrameworkElement.Language](#) , [FrameworkElement.LayoutTransform](#) , [FrameworkElement.Margin](#) ,
[FrameworkElement.MaxHeight](#) , [FrameworkElement.MaxWidth](#) , [FrameworkElement.MinHeight](#) ,
[FrameworkElement.MinWidth](#) , [FrameworkElement.Name](#) ,
[FrameworkElementOverridesDefaultStyle](#) , [FrameworkElement.Parent](#) ,
[FrameworkElement.Resources](#) , [FrameworkElement.Style](#) , [FrameworkElement.Tag](#) ,
[FrameworkElement.TemplatedParent](#) , [FrameworkElement.ToolTip](#) , [FrameworkElement.Triggers](#) ,
[FrameworkElement.UseLayoutRounding](#) , [FrameworkElement.VerticalAlignment](#) ,
[FrameworkElement.VisualChildrenCount](#) , [FrameworkElement.Width](#) ,
[FrameworkElement.ContextMenuClosing](#) , [FrameworkElement.ContextMenuOpening](#) ,
[FrameworkElement.DataContextChanged](#) , [FrameworkElement.Initialized](#) ,
[FrameworkElement.Loaded](#) , [FrameworkElement.RequestBringIntoView](#) ,
[FrameworkElement.SizeChanged](#) , [FrameworkElement.SourceUpdated](#) ,
[FrameworkElement.TargetUpdated](#) , [FrameworkElement.ToolTipClosing](#) ,
[FrameworkElement.ToolTipOpening](#) , [FrameworkElement.Unloaded](#) ,
[UIElement.AllowDropProperty](#) , [UIElement.AreAnyTouchesCapturedProperty](#) ,
[UIElement.AreAnyTouchesCapturedWithinProperty](#) , [UIElement.AreAnyTouchesDirectlyOverProperty](#) ,
[UIElement.AreAnyTouchesOverProperty](#) , [UIElement.BitmapEffectInputProperty](#) ,
[UIElement.BitmapEffectProperty](#) , [UIElement.CacheModeProperty](#) , [UIElement.ClipProperty](#) ,
[UIElement.ClipToBoundsProperty](#) , [UIElement.DragEnterEvent](#) , [UIElement.DragLeaveEvent](#) ,
[UIElement.DragOverEvent](#) , [UIElement.DropEvent](#) , [UIElement.EffectProperty](#) ,
[UIElement.FocusableProperty](#) , [UIElement.GiveFeedbackEvent](#) , [UIElement.GotFocusEvent](#) ,
[UIElement.GotKeyboardFocusEvent](#) , [UIElement.GotMouseCaptureEvent](#) ,
[UIElement.GotStylusCaptureEvent](#) , [UIElement.GotTouchCaptureEvent](#) ,
[UIElement.IsEnabledProperty](#) , [UIElement.IsFocusedProperty](#) , [UIElement.IsHitTestVisibleProperty](#) ,
[UIElement.IsKeyboardFocusedProperty](#) , [UIElement.IsKeyboardFocusWithinProperty](#) ,
[UIElement.IsManipulationEnabledProperty](#) , [UIElement.IsMouseCapturedProperty](#) ,
[UIElement.IsMouseCaptureWithinProperty](#) , [UIElement.IsMouseDirectlyOverProperty](#) ,
[UIElement.IsMouseOverProperty](#) , [UIElement.IsStylusCapturedProperty](#) ,
[UIElement.IsStylusCaptureWithinProperty](#) , [UIElement.IsStylusDirectlyOverProperty](#) ,
[UIElement.IsStylusOverProperty](#) , [UIElement.IsVisibleProperty](#) , [UIElement.KeyDownEvent](#) ,
[UIElement.KeyUpEvent](#) , [UIElement.LostFocusEvent](#) , [UIElement.LostKeyboardFocusEvent](#) ,
[UIElement.LostMouseCaptureEvent](#) , [UIElement.LostStylusCaptureEvent](#) ,
[UIElement.LostTouchCaptureEvent](#) , [UIElement.ManipulationBoundaryFeedbackEvent](#) ,
[UIElement.ManipulationCompletedEvent](#) , [UIElement.ManipulationDeltaEvent](#) ,

[UIElement.ManipulationInertiaStartingEvent](#) , [UIElement.ManipulationStartedEvent](#) ,
[UIElement.ManipulationStartingEvent](#) , [UIElement.MouseDownEvent](#) , [UIElement.MouseEnterEvent](#) ,
[UIElement.MouseLeaveEvent](#) , [UIElement.MouseLeftButtonDownEvent](#) ,
[UIElement.MouseLeftButtonUpEvent](#) , [UIElement.MouseMoveEvent](#) ,
[UIElement.MouseRightButtonDownEvent](#) , [UIElement.MouseRightButtonUpEvent](#) ,
[UIElement.MouseUpEvent](#) , [UIElement.MouseWheelEvent](#) , [UIElement.OpacityMaskProperty](#) ,
[UIElement.OpacityProperty](#) , [UIElement.PreviewDragEnterEvent](#) ,
[UIElement.PreviewDragLeaveEvent](#) , [UIElement.PreviewDragOverEvent](#) ,
[UIElement.PreviewDropEvent](#) , [UIElement.PreviewGiveFeedbackEvent](#) ,
[UIElement.PreviewGotKeyboardFocusEvent](#) , [UIElement.PreviewKeyDownEvent](#) ,
[UIElement.PreviewKeyUpEvent](#) , [UIElement.PreviewLostKeyboardFocusEvent](#) ,
[UIElement.PreviewMouseDownEvent](#) , [UIElement.PreviewMouseLeftButtonDownEvent](#) ,
[UIElement.PreviewMouseLeftButtonUpEvent](#) , [UIElement.PreviewMouseMoveEvent](#) ,
[UIElement.PreviewMouseRightButtonDownEvent](#) , [UIElement.PreviewMouseRightButtonUpEvent](#) ,
[UIElement.PreviewMouseUpEvent](#) , [UIElement.PreviewMouseWheelEvent](#) ,
[UIElement.PreviewQueryContinueDragEvent](#) , [UIElement.PreviewStylusButtonDownEvent](#) ,
[UIElement.PreviewStylusButtonUpEvent](#) , [UIElement.PreviewStylusDownEvent](#) ,
[UIElement.PreviewStylusInAirMoveEvent](#) , [UIElement.PreviewStylusInRangeEvent](#) ,
[UIElement.PreviewStylusMoveEvent](#) , [UIElement.PreviewStylusOutOfRangeEvent](#) ,
[UIElement.PreviewStylusSystemGestureEvent](#) , [UIElement.PreviewStylusUpEvent](#) ,
[UIElement.PreviewTextInputEvent](#) , [UIElement.PreviewTouchDownEvent](#) ,
[UIElement.PreviewTouchMoveEvent](#) , [UIElement.PreviewTouchUpEvent](#) ,
[UIElement.QueryContinueDragEvent](#) , [UIElement.QueryCursorEvent](#) ,
[UIElement.RenderTransformOriginProperty](#) , [UIElement.RenderTransformProperty](#) ,
[UIElement.SnapsToDevicePixelsProperty](#) , [UIElement.StylusButtonDownEvent](#) ,
[UIElement.StylusButtonUpEvent](#) , [UIElement.StylusDownEvent](#) , [UIElement.StylusEnterEvent](#) ,
[UIElement.StylusInAirMoveEvent](#) , [UIElement.StylusInRangeEvent](#) , [UIElement.StylusLeaveEvent](#) ,
[UIElement.StylusMoveEvent](#) , [UIElement.StylusOutOfRangeEvent](#) ,
[UIElement.StylusSystemGestureEvent](#) , [UIElement.StylusUpEvent](#) , [UIElement.TextInputEvent](#) ,
[UIElement.TouchDownEvent](#) , [UIElement.TouchEnterEvent](#) , [UIElement.TouchLeaveEvent](#) ,
[UIElement.TouchMoveEvent](#) , [UIElement.TouchUpEvent](#) , [UIElement.UidProperty](#) ,
[UIElement.VisibilityProperty](#) , [UIElement.AddHandler\(RoutedEvent, Delegate\)](#) ,
[UIElement.AddHandler\(RoutedEvent, Delegate, bool\)](#) ,
[UIElement.AddToEventRoute\(EventRoute, RoutedEventArgs\)](#) ,
[UIElement.ApplyAnimationClock\(DependencyProperty, AnimationClock\)](#) ,
[UIElement.ApplyAnimationClock\(DependencyProperty, AnimationClock, HandoffBehavior\)](#) ,
[UIElement.Arrange\(Rect\)](#) , [UIElement.BeginAnimation\(DependencyProperty, AnimationTimeline\)](#) ,
[UIElement.BeginAnimation\(DependencyProperty, AnimationTimeline, HandoffBehavior\)](#) ,
[UIElement.CaptureMouse\(\)](#) , [UIElement.CaptureStylus\(\)](#) , [UIElement.CaptureTouch\(TouchDevice\)](#) ,
[UIElement.Focus\(\)](#) , [UIElement.GetAnimationBaseValue\(DependencyProperty\)](#) ,

[UIElement.HitTestCore\(GeometryHitTestParameters\)](#) , [UIElement.HitTestCore\(PointHitTestParameters\)](#) ,
[UIElement.InputHitTest\(Point\)](#) , [UIElement.InvalidateArrange\(\)](#) , [UIElement.InvalidateMeasure\(\)](#) ,
[UIElement.InvalidateVisual\(\)](#) , [UIElement.Measure\(Size\)](#) ,
[UIElement.OnAccessKey\(AccessKeyEventArgs\)](#) , [UIElement.OnChildDesiredSizeChanged\(UIElement\)](#) ,
[UIElement.OnDragEnter\(DragEventArgs\)](#) , [UIElement.OnDragLeave\(DragEventArgs\)](#) ,
[UIElement.OnDragOver\(DragEventArgs\)](#) , [UIElement.OnDrop\(DragEventArgs\)](#) ,
[UIElement.OnGiveFeedback\(GiveFeedbackEventArgs\)](#) ,
[UIElement.OnGotKeyboardFocus\(KeyboardFocusChangedEventArgs\)](#) ,
[UIElement.OnGotMouseCapture\(MouseEventArgs\)](#) ,
[UIElement.OnGotStylusCapture\(StylusEventArgs\)](#) , [UIElement.OnGotTouchCapture\(TouchEventArgs\)](#) ,
[UIElement.OnIsKeyboardFocusedChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsKeyboardFocusWithinChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsMouseCapturedChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsMouseCaptureWithinChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsMouseDirectlyOverChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsStylusCapturedChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsStylusCaptureWithinChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsStylusDirectlyOverChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnKeyDown\(KeyEventArgs\)](#) , [UIElement.OnKeyUp\(KeyEventArgs\)](#) ,
[UIElement.OnLostFocus\(RoutedEventArgs\)](#) ,
[UIElement.OnLostKeyboardFocus\(KeyboardFocusChangedEventArgs\)](#) ,
[UIElement.OnLostMouseCapture\(MouseEventArgs\)](#) ,
[UIElement.OnLostStylusCapture\(StylusEventArgs\)](#) , [UIElement.OnLostTouchCapture\(TouchEventArgs\)](#) ,
[UIElement.OnManipulationCompleted\(ManipulationCompletedEventArgs\)](#) ,
[UIElement.OnManipulationDelta\(ManipulationDeltaEventArgs\)](#) ,
[UIElement.OnManipulationInertiaStarting\(ManipulationInertiaStartingEventArgs\)](#) ,
[UIElement.OnManipulationStarted\(ManipulationStartedEventArgs\)](#) ,
[UIElement.OnManipulationStarting\(ManipulationStartingEventArgs\)](#) ,
[UIElement.OnMouseDown\(MouseButtonEventArgs\)](#) , [UIElement.OnMouseEnter\(MouseEventArgs\)](#) ,
[UIElement.OnMouseLeave\(MouseEventArgs\)](#) ,
[UIElement.OnMouseLeftButtonDown\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseLeftButtonUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseMove\(MouseEventArgs\)](#) ,
[UIElement.OnMouseRightButtonDown\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseRightButtonUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseWheel\(MouseWheelEventArgs\)](#) ,
[UIElement.OnPreviewDragEnter\(DragEventArgs\)](#) , [UIElement.OnPreviewDragLeave\(DragEventArgs\)](#) ,
[UIElement.OnPreviewDragOver\(DragEventArgs\)](#) , [UIElement.OnPreviewDrop\(DragEventArgs\)](#) ,
[UIElement.OnPreviewGiveFeedback\(GiveFeedbackEventArgs\)](#) ,

[UIElement.OnPreviewGotKeyboardFocus\(KeyboardFocusChangedEventArgs\)](#) ,
[UIElement.OnPreviewKeyDown\(KeyEventArgs\)](#) , [UIElement.OnPreviewKeyUp\(KeyEventArgs\)](#) ,
[UIElement.OnPreviewLostKeyboardFocus\(KeyboardFocusChangedEventArgs\)](#) ,
[UIElement.OnPreviewMouseDown\(MouseButtonEventArgs\)](#) ,
[UIElement.OnPreviewMouseLeftButtonDown\(MouseButtonEventArgs\)](#) ,
[UIElement.OnPreviewMouseLeftButtonUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnPreviewMouseMove\(MouseEventArgs\)](#) ,
[UIElement.OnPreviewMouseRightButtonDown\(MouseButtonEventArgs\)](#) ,
[UIElement.OnPreviewMouseRightButtonUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnPreviewMouseUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnPreviewMouseWheel\(MouseWheelEventArgs\)](#) ,
[UIElement.OnPreviewQueryContinueDrag\(QueryContinueDragEventArgs\)](#) ,
[UIElement.OnPreviewStylusButtonDown\(StylusButtonEventArgs\)](#) ,
[UIElement.OnPreviewStylusButtonUp\(StylusButtonEventArgs\)](#) ,
[UIElement.OnPreviewStylusDown\(StylusDownEventArgs\)](#) ,
[UIElement.OnPreviewStylusInAirMove\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewStylusInRange\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewStylusMove\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewStylusOutOfRange\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewStylusSystemGesture\(StylusSystemGestureEventArgs\)](#) ,
[UIElement.OnPreviewStylusUp\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewTextInput\(TextCompositionEventArgs\)](#) ,
[UIElement.OnPreviewTouchDown\(TouchEventArgs\)](#) ,
[UIElement.OnPreviewTouchMove\(TouchEventArgs\)](#) , [UIElement.OnPreviewTouchUp\(TouchEventArgs\)](#) ,
[UIElement.OnQueryContinueDrag\(QueryContinueDragEventArgs\)](#) ,
[UIElement.OnQueryCursor\(QueryCursorEventArgs\)](#) , [UIElement.OnRender\(DrawingContext\)](#) ,
[UIElement.OnStylusButtonDown\(StylusButtonEventArgs\)](#) ,
[UIElement.OnStylusButtonUp\(StylusButtonEventArgs\)](#) ,
[UIElement.OnStylusDown\(StylusDownEventArgs\)](#) , [UIElement.OnStylusEnter\(StylusEventArgs\)](#) ,
[UIElement.OnStylusInAirMove\(StylusEventArgs\)](#) , [UIElement.OnStylusInRange\(StylusEventArgs\)](#) ,
[UIElement.OnStylusLeave\(StylusEventArgs\)](#) , [UIElement.OnStylusMove\(StylusEventArgs\)](#) ,
[UIElement.OnStylusOutOfRange\(StylusEventArgs\)](#) ,
[UIElement.OnStylusSystemGesture\(StylusSystemGestureEventArgs\)](#) ,
[UIElement.OnStylusUp\(StylusEventArgs\)](#) , [UIElement.OnTextInput\(TextCompositionEventArgs\)](#) ,
[UIElement.OnTouchDown\(TouchEventArgs\)](#) , [UIElement.OnTouchEnter\(TouchEventArgs\)](#) ,
[UIElement.OnTouchLeave\(TouchEventArgs\)](#) , [UIElement.OnTouchMove\(TouchEventArgs\)](#) ,
[UIElement.OnTouchUp\(TouchEventArgs\)](#) , [UIElement.RaiseEvent\(RoutedEventArgs\)](#) ,
[UIElement.ReleaseAllTouchCaptures\(\)](#) , [UIElement.ReleaseMouseCapture\(\)](#) ,
[UIElement.ReleaseStylusCapture\(\)](#) , [UIElement.ReleaseTouchCapture\(TouchDevice\)](#) ,
[UIElement.RemoveHandler\(RoutedEvent, Delegate\)](#) , [UIElement.TranslatePoint\(Point, UIElement\)](#) ,

[UIElement.UpdateLayout\(\)](#) , [UIElement.AllowDrop](#) , [UIElement.AreAnyTouchesCaptured](#) ,
[UIElement.AreAnyTouchesCapturedWithin](#) , [UIElement.AreAnyTouchesDirectlyOver](#) ,
[UIElement.AreAnyTouchesOver](#) , [UIElement.BitmapEffect](#) , [UIElement.BitmapEffectInput](#) ,
[UIElement.CacheMode](#) , [UIElement.Clip](#) , [UIElement.ClipToBounds](#) ,
[UIElement.CommandBindings](#) , [UIElement.DesiredSize](#) , [UIElement.Effect](#) , [UIElement.Focusable](#) ,
[UIElement.HasAnimatedProperties](#) , [UIElement.HasEffectiveKeyboardFocus](#) ,
[UIElement.InputBindings](#) , [UIElement.IsArrangeValid](#) , [UIElement.IsEnabled](#) ,
[UIElement.IsEnabledCore](#) , [UIElement.IsFocused](#) , [UIElement.IsHitTestVisible](#) ,
[UIElement.IsInputMethodEnabled](#) , [UIElement.IsKeyboardFocused](#) ,
[UIElement.IsKeyboardFocusWithin](#) , [UIElement.IsManipulationEnabled](#) , [UIElement.IsMeasureValid](#) ,
[UIElement.IsMouseCaptured](#) , [UIElement.IsMouseCaptureWithin](#) , [UIElement.IsMouseDirectlyOver](#) ,
[UIElement.IsMouseOver](#) , [UIElement.IsStylusCaptured](#) , [UIElement.IsStylusCaptureWithin](#) ,
[UIElement.IsStylusDirectlyOver](#) , [UIElement.IsStylusOver](#) , [UIElement.IsVisible](#) , [UIElement.Opacity](#) ,
[UIElement.OpacityMask](#) , [UIElement.PersistId](#) , [UIElement.RenderSize](#) ,
[UIElement.RenderTransform](#) , [UIElement.RenderTransformOrigin](#) , [UIElement.SnapsToDevicePixels](#) ,
[UIElement.StylusPlugIns](#) , [UIElement.TouchesCaptured](#) , [UIElement.TouchesCapturedWithin](#) ,
[UIElement.TouchesDirectlyOver](#) , [UIElement.TouchesOver](#) , [UIElement.Uid](#) , [UIElement.Visibility](#) ,
[UIElement.DragEnter](#) , [UIElement.DragLeave](#) , [UIElement.DragOver](#) , [UIElement.Drop](#) ,
[UIElement.FocusableChanged](#) , [UIElement.GiveFeedback](#) , [UIElement.GotFocus](#) ,
[UIElement.GotKeyboardFocus](#) , [UIElement.GotMouseCapture](#) , [UIElement.GotStylusCapture](#) ,
[UIElement.GotTouchCapture](#) , [UIElement.IsEnabledChanged](#) , [UIElement.IsHitTestVisibleChanged](#) ,
[UIElement.IsKeyboardFocusedChanged](#) , [UIElement.IsKeyboardFocusWithinChanged](#) ,
[UIElement.IsMouseCapturedChanged](#) , [UIElement.IsMouseCaptureWithinChanged](#) ,
[UIElement.IsMouseDirectlyOverChanged](#) , [UIElement.IsStylusCapturedChanged](#) ,
[UIElement.IsStylusCaptureWithinChanged](#) , [UIElement.IsStylusDirectlyOverChanged](#) ,
[UIElement.IsVisibleChanged](#) , [UIElement.KeyDown](#) , [UIElement.KeyUp](#) ,
[UIElement.LayoutUpdated](#) , [UIElement.LostFocus](#) , [UIElement.LostKeyboardFocus](#) ,
[UIElement.LostMouseCapture](#) , [UIElement.LostStylusCapture](#) , [UIElement.LostTouchCapture](#) ,
[UIElement.ManipulationBoundaryFeedback](#) , [UIElement.ManipulationCompleted](#) ,
[UIElement.ManipulationDelta](#) , [UIElement.ManipulationInertiaStarting](#) ,
[UIElement.ManipulationStarted](#) , [UIElement.ManipulationStarting](#) , [UIElement.MouseDown](#) ,
[UIElement.MouseEnter](#) , [UIElement.MouseLeave](#) , [UIElement.MouseLeftButtonDown](#) ,
[UIElement.MouseLeftButtonUp](#) , [UIElement.MouseMove](#) , [UIElement.MouseRightButtonDown](#) ,
[UIElement.MouseRightButtonUp](#) , [UIElement.MouseUp](#) , [UIElement.MouseWheel](#) ,
[UIElement.PreviewDragEnter](#) , [UIElement.PreviewDragLeave](#) , [UIElement.PreviewDragOver](#) ,
[UIElement.PreviewDrop](#) , [UIElement.PreviewGiveFeedback](#) , [UIElement.PreviewGotKeyboardFocus](#) ,
[UIElement.PreviewKeyDown](#) , [UIElement.PreviewKeyUp](#) , [UIElement.PreviewLostKeyboardFocus](#) ,
[UIElement.PreviewMouseDown](#) , [UIElement.PreviewMouseLeftButtonDown](#) ,
[UIElement.PreviewMouseLeftButtonUp](#) , [UIElement.PreviewMouseMove](#) ,
[UIElement.PreviewMouseRightButtonDown](#) , [UIElement.PreviewMouseRightButtonUp](#) ,

[UIElement.PreviewMouseUp](#) , [UIElement.PreviewMouseWheel](#) ,
[UIElement.PreviewQueryContinueDrag](#) , [UIElement.PreviewStylusButtonDown](#) ,
[UIElement.PreviewStylusButtonUp](#) , [UIElement.PreviewStylusDown](#) ,
[UIElement.PreviewStylusInAirMove](#) , [UIElement.PreviewStylusInRange](#) ,
[UIElement.PreviewStylusMove](#) , [UIElement.PreviewStylusOutOfRange](#) ,
[UIElement.PreviewStylusSystemGesture](#) , [UIElement.PreviewStylusUp](#) , [UIElement.PreviewTextInput](#) ,
[UIElement.PreviewTouchDown](#) , [UIElement.PreviewTouchMove](#) , [UIElement.PreviewTouchUp](#) ,
[UIElement.QueryContinueDrag](#) , [UIElement.QueryCursor](#) , [UIElement.StylusButtonDown](#) ,
[UIElement.StylusButtonUp](#) , [UIElement.StylusDown](#) , [UIElement.StylusEnter](#) ,
[UIElement.StylusInAirMove](#) , [UIElement.StylusInRange](#) , [UIElement.StylusLeave](#) ,
[UIElement.StylusMove](#) , [UIElement.StylusOutOfRange](#) , [UIElement.StylusSystemGesture](#) ,
[UIElement.StylusUp](#) , [UIElement.TextInput](#) , [UIElement.TouchDown](#) , [UIElement.TouchEnter](#) ,
[UIElement.TouchLeave](#) , [UIElement.TouchMove](#) , [UIElement.TouchUp](#) ,
[Visual.AddVisualChild\(Visual\)](#) , [Visual.FindCommonVisualAncestor\(DependencyObject\)](#) ,
[Visual.IsAncestorOf\(DependencyObject\)](#) , [Visual.IsDescendantOf\(DependencyObject\)](#) ,
[Visual.PointFromScreen\(Point\)](#) , [Visual.PointToScreen\(Point\)](#) , [Visual.RemoveVisualChild\(Visual\)](#) ,
[Visual.TransformToAncestor\(Visual3D\)](#) , [Visual.TransformToAncestor\(Visual\)](#) ,
[Visual.TransformToDescendant\(Visual\)](#) , [Visual.TransformToVisual\(Visual\)](#) , [Visual.VisualBitmapEffect](#) ,
[Visual.VisualBitmapEffectInput](#) , [Visual.VisualBitmapScalingMode](#) , [Visual.VisualCacheMode](#) ,
[Visual.VisualClearTypeHint](#) , [Visual.VisualClip](#) , [Visual.VisualEdgeMode](#) , [Visual.VisualEffect](#) ,
[Visual.VisualOffset](#) , [Visual.VisualOpacity](#) , [Visual.VisualOpacityMask](#) , [Visual.VisualParent](#) ,
[Visual.VisualScrollableAreaClip](#) , [Visual.VisualTextHintingMode](#) , [Visual.VisualTextRenderingMode](#) ,
[Visual.VisualTransform](#) , [Visual.VisualXSnappingGuidelines](#) , [Visual.VisualYSnappingGuidelines](#) ,
[DependencyObject.ClearValue\(DependencyProperty\)](#) ,
[DependencyObject.ClearValue\(DependencyPropertyKey\)](#) ,
[DependencyObject.CoerceValue\(DependencyProperty\)](#) , [DependencyObject.Equals\(object\)](#) ,
[DependencyObject.GetHashCode\(\)](#) , [DependencyObject.GetLocalValueEnumerator\(\)](#) ,
[DependencyObject.GetValue\(DependencyProperty\)](#) ,
[DependencyObject.InvalidateProperty\(DependencyProperty\)](#) ,
[DependencyObject.ReadLocalValue\(DependencyProperty\)](#) ,
[DependencyObject.SetCurrentValue\(DependencyProperty, object\)](#) ,
[DependencyObject.SetValue\(DependencyProperty, object\)](#) ,
[DependencyObject.SetValue\(DependencyPropertyKey, object\)](#) ,
[DependencyObject.ShouldSerializeProperty\(DependencyProperty\)](#) ,
[DependencyObject.DependencyObjectType](#) , [DependencyObject.IsSealed](#) ,
[DispatcherObject.Dispatcher](#) , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

CanvasChart(string, ref Canvas, bool)

Constructor

```
public CanvasChart(string windowTitle, ref Canvas inCanvas, bool useLinesBool = true)
```

Parameters

windowTitle [string](#)

title for the window

inCanvas [Canvas](#)

to use 'Canvas'

useLinesBool [bool](#)

show the dots connected with lines

Fields

titelText

titel of the coordinate system

```
public string titelText
```

Field Value

[string](#)

version

created on: 08.07.2023

last edit: 20.09.24

```
public Version version
```

Field Value

[Version ↗](#)

Methods

DataAdd(List<long>, List<double>)

Saves the dataset in two lists (X, Y).

Used with the error data.

```
public void DataAdd(List<long> inX, List<double> inY)
```

Parameters

inX [List ↗ <long ↗ >](#)

field with the coordinates for X

inY [List ↗ <double ↗ >](#)

field with the coordinates for Y

DataAdd(double[], double[])

Saves the dataset in two lists (X, Y).

Used with normal data.

```
public void DataAdd(double[] inX, double[] inY)
```

Parameters

inX [double ↗ \[\]](#)

field with the coordinates for X

inY [double ↗ \[\]](#)

field with the coordinates for Y

DataClear()

Clears the datalists.

```
public void DataClear()
```

GetCanvasWindowXpos(double)

Calculates from the logical datavalue the real x-position in the chart.

```
public double GetCanvasWindowXpos(double xValue)
```

Parameters

xValue [double](#)

one of the datasets value

Returns

[double](#)

the real x-position in the 'Canvas'

GetCanvasWindowYpos(double)

Calculates from the logical datavalue the real x-position in the chart.

```
public double GetCanvasWindowYpos(double yValue)
```

Parameters

yValue [double](#)

one of the datasets value

Returns

double ↗

the real y-position in the 'Canvas'

GetFieldsLastElements(List<double[]>)

Delivers the wanted last elements of a list containing double-fields.

```
public List<double[]> GetFieldsLastElements(List<double[]> inList)
```

Parameters

inList List ↗ <double ↗ []>

input list

Returns

List ↗ <double ↗ []>

new list with the number of elements

GetLastElements(double[])

Dependent on 'showNoOfData' the lessered length of the inputfield will be given back.

```
public double[] GetLastElements(double[] inArray)
```

Parameters

inArray double ↗ []

the inputarray

Returns

double ↗ []

the shortened field

GetXpos(double)

Calculates from the logical datavalue the position in the diagramm for the x-position.

```
public double GetXpos(double xValue)
```

Parameters

xValue [double](#)

a value of the dataset

Returns

[double](#)

the real x-position on the 'Canvas'

GetYpos(double)

Calculates from the logical datavalue the position in the diagramm for the y-position.

```
public double GetYpos(double yValue)
```

Parameters

yValue [double](#)

a value of the dataset

Returns

[double](#)

the real y-position on the 'Canvas'

SetCanvasWindowLabel(double, double, string, int, bool)

Helperfunction for 'canvasWindowCanvas' setting a label on the given coordinates. Can inscribe chart objects.

```
public void SetCanvasWindowLabel(double xPos, double yPos, string text, int fontSize = 10,  
bool bold = false)
```

Parameters

xPos [double](#)

distance from the left border

yPos [double](#)

distance from the top

text [string](#)

the message

fontSize [int](#)

standard size is 10

bold [bool](#)

SetCanvasWindowLine(double, double, double, double, int, int)

Helperfunction putting a line into 'canvasWindowCanvas'.

```
public void SetCanvasWindowLine(double x1, double y1, double x2, double y2, int thickness,  
int brushColor)
```

Parameters

x1 [double](#)

from x-coordinte

y1 [double](#)

from y-coordinate

x2 [double](#)

to x-coordinate

y2 [double](#)

to y-coordinate

thickness [int](#)

standard broadness is 2

brushColor [int](#)

color of the brush

SetCanvasWindowLineStroked(double, double, double, double, int, int)

Helperfunction putting a stroked line into 'canvasWindowCanvas'.

```
public void SetCanvasWindowLineStroked(double x1, double y1, double x2, double y2, int thickness, int brushColor)
```

Parameters

x1 [double](#)

from x-coordinte

y1 [double](#)

from y-coordinate

x2 [double](#)

to x-coordinate

y2 [double](#)

to y-coordinate

thickness [int](#)

standard broadness is 2

brushColor [int](#)

color of the brush

SetCanvasWindowXaxisLabel(double)

Sets the x-axle label.

```
public void SetCanvasWindowXaxisLabel(double xPosition)
```

Parameters

xPosition [double](#)

x-position of the text

SetCanvasWindowYaxisLabel(double)

Sets the y-axle label.

```
public void SetCanvasWindowYaxisLabel(double yPosition)
```

Parameters

yPosition [double](#)

y-position of the text

SetDrawAreaLabel(double, double, string, int, bool)

Helpfunction for the 'drawArea' setting a label onto the given position. Can inscribe the chart.

```
public void SetDrawAreaLabel(double xPos, double yPos, string text, int fontSize = 10, bool bold = false)
```

Parameters

xPos [double](#)

distance from the left border

yPos [double](#)

distance from the upper border

text [string](#)

the string

fontSize [int](#)

the fonts size

bold [bool](#)

bold?

SetDrawAreaLine(double, double, double, double, int, int)

Helpfunction setting a line into the 'drawArea'.

```
public void SetDrawAreaLine(double x1, double y1, double x2, double y2, int thickness,  
int brushNumber)
```

Parameters

x1 [double](#)

from the x-coordinate

y1 [double](#)

from the y-coordinate

x2 [double](#)

to the x-coordinate

y2 [double](#)

to the y-coordinate

thickness [int ↗](#)

linebroadness

brushNumber [int ↗](#)

number of the brush

SetDrawAreaLineStroked(double, double, double, double, int, int)

Helpfunction setting a stroked line into the 'drawArea'.

```
public void SetDrawAreaLineStroked(double x1, double y1, double x2, double y2, int  
thickness, int brushNumber)
```

Parameters

x1 [double ↗](#)

from the x-coordinate

y1 [double ↗](#)

from the y-coordinate

x2 [double ↗](#)

to the x-coordinate

y2 [double ↗](#)

to the y-coordinate

thickness [int ↗](#)

linebroadness

brushNumber [int ↗](#)

number of the brush

SetShowNoOfData(int)

Set this value to only show the last elements of the dataset in the chart.

0 show all the data.

```
public void SetShowNoOfData(int inNo = 0)
```

Parameters

inNo [int](#)

the new value 'showNoOfData'

SetXaxisLabel(double)

Sets the label of the x-axis.

```
public void SetXaxisLabel(double xPosition)
```

Parameters

xPosition [double](#)

position on the 'Canvas'

SetYaxisLabel(double)

Sets the label of the y-axis.

```
public void SetYaxisLabel(double yPosition)
```

Parameters

yPosition [double](#)

position on the 'Canvas'

ShowChart()

Shows the coordinate system scaled to the dataset
including the 'showNoOfData' last elements. You need more than one datapoint or nothing is shown!

The next first use of LINQ - an achievement for me.

```
public void ShowChart()
```

ShowWindow()

Shows the data in the window ('CanvasWindow')

```
public void ShowWindow()
```

Class CanvasTopic

Namespace: [MatrixFFN.Tools](#)

Assembly: MatrixFFN.dll

Shows the nets structure graphical in the given 'Canvas'..

Additionally you can open the 'CanvasWindow' to see a bigger 'Canvas' with the diagramm.

```
public class CanvasTopic : CanvasWindow, IAnimatable, ISupportInitialize,
IFrameworkInputElement, IInputElement, IQueryAmbient, IAddChild, IComponentConnector
```

Inheritance

```
object ↗ ← DispatcherObject ↗ ← DependencyObject ↗ ← Visual ↗ ← UIElement ↗ ←
FrameworkElement ↗ ← Control ↗ ← ContentControl ↗ ← Window ↗ ← CanvasWindow ← CanvasTopic
```

Implements

```
IAnimatable ↗ , ISupportInitialize ↗ , IFrameworkInputElement ↗ , IInputElement ↗ , IQueryAmbient ↗ ,
IAddChild ↗ , IComponentConnector ↗
```

Inherited Members

```
CanvasWindow.isNowToEnd , CanvasWindow.InitializeComponent() ,
WindowAllowsTransparencyProperty ↗ , Window.DpiChangedEvent ↗ , Window.IconProperty ↗ ,
Window.IsActiveProperty ↗ , Window.LeftProperty ↗ , Window.ResizeModeProperty ↗ ,
Window>ShowActivatedProperty ↗ , Window>ShowInTaskbarProperty ↗ ,
Window.SizeToContentProperty ↗ , Window.TaskBarItemInfoProperty ↗ , Window.TitleProperty ↗ ,
Window.TopmostProperty ↗ , Window.TopProperty ↗ , Window.WindowStateProperty ↗ ,
Window.WindowStyleProperty ↗ , Window.Activate() ↗ , Window.ArrangeOverride(Size) ↗ ,
Window.Close() ↗ , Window.DragMove() ↗ , Window.GetWindow(DependencyObject) ↗ ,
Window.Hide() ↗ , Window.MeasureOverride(Size) ↗ , Window.OnActivated(EventArgs) ↗ ,
Window.OnClosed(EventArgs) ↗ , Window.OnClosing(CancelEventArgs) ↗ ,
Window.OnContentChanged(object, object) ↗ , Window.OnContentRendered(EventArgs) ↗ ,
Window.OnCreateAutomationPeer() ↗ , Window.OnDeactivated(EventArgs) ↗ ,
Window.OnDpiChanged(DpiScale, DpiScale) ↗ , Window.OnLocationChanged(EventArgs) ↗ ,
Window.OnManipulationBoundaryFeedback(ManipulationBoundaryFeedbackEventArgs) ↗ ,
Window.OnSourceInitialized(EventArgs) ↗ , Window.OnStateChanged(EventArgs) ↗ ,
Window.OnVisualChildrenChanged(DependencyObject, DependencyObject) ↗ ,
Window.OnVisualParentChanged(DependencyObject) ↗ , Window>Show() ↗ , Window>ShowDialog() ↗ ,
WindowAllowsTransparency ↗ , Window.DialogResult ↗ , Window.Icon ↗ , Window.IsActive ↗ ,
Window.Left ↗ , Window.LogicalChildren ↗ , Window.OwnedWindows ↗ , Window.Owner ↗ ,
```

[Window.ResizeMode](#) , [Window.RestoreBounds](#) , [Window>ShowActivated](#) ,
[Window>ShowInTaskbar](#) , [Window>SizeToContent](#) , [Window>TaskBarItemInfo](#) , [Window>Title](#) ,
[Window>Top](#) , [Window>Topmost](#) , [Window>WindowStartupLocation](#) , [Window>WindowState](#) ,
[Window>WindowStyle](#) , [Window>Activated](#) , [Window>Closed](#) , [Window>Closing](#) ,
[Window>ContentRendered](#) , [Window>Deactivated](#) , [Window>DpiChanged](#) ,
[Window>LocationChanged](#) , [Window>SourcesInitialized](#) , [Window>StateChanged](#) ,
[ContentControl>ContentProperty](#) , [ContentControl>ContentStringFormatProperty](#) ,
[ContentControl>ContentTemplateProperty](#) , [ContentControl>ContentTemplateSelectorProperty](#) ,
[ContentControl>HasContentProperty](#) , [ContentControl>AddChild\(object\)](#) ,
[ContentControl>AddText\(string\)](#) , [ContentControl>OnContentStringFormatChanged\(string, string\)](#) ,
[ContentControl>OnContentTemplateChanged\(DataTemplate, DataTemplate\)](#) ,
[ContentControl>OnContentTemplateSelectorChanged\(DataTemplateSelector, DataTemplateSelector\)](#) ,
[ContentControl>Content](#) , [ContentControl>ContentStringFormat](#) , [ContentControl>ContentTemplate](#) ,
[ContentControl>ContentTemplateSelector](#) , [ContentControl>HasContent](#) ,
[Control>BackgroundProperty](#) , [Control>BorderBrushProperty](#) , [Control>BorderThicknessProperty](#) ,
[Control>FontFamilyProperty](#) , [Control>FontSizeProperty](#) , [Control>FontStretchProperty](#) ,
[Control>FontStyleProperty](#) , [Control>FontWeightProperty](#) , [Control>ForegroundProperty](#) ,
[Control>HorizontalContentAlignmentProperty](#) , [Control>IsTabStopProperty](#) ,
[Control>MouseDoubleClickEvent](#) , [Control>PaddingProperty](#) ,
[Control>PreviewMouseDoubleClickEvent](#) , [Control>TabIndexProperty](#) , [Control>TemplateProperty](#) ,
[Control>VerticalContentAlignmentProperty](#) , [Control>OnMouseDoubleClick\(MouseEventArgs\)](#) ,
[Control>OnPreviewMouseDoubleClick\(MouseEventArgs\)](#) ,
[Control>OnTemplateChanged\(ControlTemplate, ControlTemplate\)](#) , [Control>ToString\(\)](#) ,
[Control>Background](#) , [Control>BorderBrush](#) , [Control>BorderThickness](#) , [Control>FontFamily](#) ,
[Control>FontSize](#) , [Control>FontStretch](#) , [Control>FontStyle](#) , [Control>FontWeight](#) ,
[Control>Foreground](#) , [Control>HandlesScrolling](#) , [Control>HorizontalContentAlignment](#) ,
[Control>IsTabStop](#) , [Control>Padding](#) , [Control>TabIndex](#) , [Control>Template](#) ,
[Control>VerticalContentAlignment](#) , [Control>MouseDoubleClick](#) ,
[Control>PreviewMouseDoubleClick](#) , [FrameworkElement>ActualHeightProperty](#) ,
[FrameworkElement>ActualWidthProperty](#) , [FrameworkElement>BindingGroupProperty](#) ,
[FrameworkElement>ContextMenuClosingEvent](#) , [FrameworkElement>ContextMenuOpeningEvent](#) ,
[FrameworkElement>ContextMenuProperty](#) , [FrameworkElement>CursorProperty](#) ,
[FrameworkElement>DataContextProperty](#) , [FrameworkElement>DefaultStyleKeyProperty](#) ,
[FrameworkElement>FlowDirectionProperty](#) , [FrameworkElement>FocusVisualStyleProperty](#) ,
[FrameworkElement>ForceCursorProperty](#) , [FrameworkElement>HeightProperty](#) ,
[FrameworkElement>HorizontalAlignmentProperty](#) , [FrameworkElement>InputScopeProperty](#) ,
[FrameworkElement>LanguageProperty](#) , [FrameworkElement>LayoutTransformProperty](#) ,
[FrameworkElement>LoadedEvent](#) , [FrameworkElement>MarginProperty](#) ,
[FrameworkElement>MaxHeightProperty](#) , [FrameworkElement>MaxWidthProperty](#) ,
[FrameworkElement>MinHeightProperty](#) , [FrameworkElement>MinWidthProperty](#) ,

[FrameworkElement.NameProperty](#) , [FrameworkElement.OverridesDefaultStyleProperty](#) ,
[FrameworkElement.RequestBringIntoViewEvent](#) , [FrameworkElement.SizeChangedEvent](#) ,
[FrameworkElement.StyleProperty](#) , [FrameworkElement.TagProperty](#) ,
[FrameworkElement.ToolTipClosingEvent](#) , [FrameworkElement.ToolTipOpeningEvent](#) ,
[FrameworkElement.ToolTipProperty](#) , [FrameworkElement.UnloadedEvent](#) ,
[FrameworkElement.UseLayoutRoundingProperty](#) , [FrameworkElement.VerticalAlignmentProperty](#) ,
[FrameworkElement.WidthProperty](#) , [FrameworkElement.AddLogicalChild\(object\)](#) ,
[FrameworkElement.ApplyTemplate\(\)](#) , [FrameworkElement.ArrangeCore\(Rect\)](#) ,
[FrameworkElement.BeginInit\(\)](#) , [FrameworkElement.BeginStoryboard\(Storyboard\)](#) ,
[FrameworkElement.BeginStoryboard\(Storyboard, HandoffBehavior\)](#) ,
[FrameworkElement.BeginStoryboard\(Storyboard, HandoffBehavior, bool\)](#) ,
[FrameworkElement.BringIntoView\(\)](#) , [FrameworkElement.BringIntoView\(Rect\)](#) ,
[FrameworkElement.EndInit\(\)](#) , [FrameworkElement.FindName\(string\)](#) ,
[FrameworkElement.FindResource\(object\)](#) ,
[FrameworkElement.GetBindingExpression\(DependencyProperty\)](#) ,
[FrameworkElement.GetFlowDirection\(DependencyObject\)](#) , [FrameworkElement.GetLayoutClip\(Size\)](#) ,
[FrameworkElement.GetTemplateChild\(string\)](#) , [FrameworkElement.GetUIParentCore\(\)](#) ,
[FrameworkElement.GetVisualChild\(int\)](#) , [FrameworkElement.MeasureCore\(Size\)](#) ,
[FrameworkElement.MoveFocus\(TraversalRequest\)](#) , [FrameworkElement.OnApplyTemplate\(\)](#) ,
[FrameworkElement.OnContextMenuClosing\(ContextMenuEventArgs\)](#) ,
[FrameworkElement.OnContextMenuOpening\(ContextMenuEventArgs\)](#) ,
[FrameworkElement.OnGotFocus\(RoutedEventArgs\)](#) , [FrameworkElement.OnInitialized\(EventArgs\)](#) ,
[FrameworkElement.OnPropertyChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[FrameworkElement.OnRenderSizeChanged\(SizeChangedEventArgs\)](#) ,
[FrameworkElement.OnStyleChanged\(Style, Style\)](#) ,
[FrameworkElement.OnToolTipClosing\(ToolTipEventArgs\)](#) ,
[FrameworkElement.OnToolTipOpening\(ToolTipEventArgs\)](#) ,
[FrameworkElement.ParentLayoutInvalidated\(UIElement\)](#) ,
[FrameworkElement.PredictFocus\(FocusNavigationDirection\)](#) ,
[FrameworkElement.RegisterName\(string, object\)](#) , [FrameworkElement.RemoveLogicalChild\(object\)](#) ,
[FrameworkElement.SetBinding\(DependencyProperty, string\)](#) ,
[FrameworkElement.SetBinding\(DependencyProperty, BindingBase\)](#) ,
[FrameworkElement.SetFlowDirection\(DependencyObject, FlowDirection\)](#) ,
[FrameworkElement.SetResourceReference\(DependencyProperty, object\)](#) ,
[FrameworkElement.TryFindResource\(object\)](#) , [FrameworkElement.UnregisterName\(string\)](#) ,
[FrameworkElement.UpdateDefaultStyle\(\)](#) , [FrameworkElement.ActualHeight](#) ,
[FrameworkElement.ActualWidth](#) , [FrameworkElement.BindingGroup](#) ,
[FrameworkElement.ContextMenu](#) , [FrameworkElement.Cursor](#) , [FrameworkElement.DataContext](#) ,
[FrameworkElement.DefaultStyleKey](#) , [FrameworkElement.FlowDirection](#) ,
[FrameworkElement.FocusVisualStyle](#) , [FrameworkElement.ForceCursor](#) , [FrameworkElement.Height](#) ,

[FrameworkElement.HorizontalAlignment](#) , [FrameworkElement.InheritanceBehavior](#) ,
[FrameworkElement.InputScope](#) , [FrameworkElement.IsInitialized](#) , [FrameworkElement.IsLoaded](#) ,
[FrameworkElement.Language](#) , [FrameworkElement.LayoutTransform](#) , [FrameworkElement.Margin](#) ,
[FrameworkElement.MaxHeight](#) , [FrameworkElement.MaxWidth](#) , [FrameworkElement.MinHeight](#) ,
[FrameworkElement.MinWidth](#) , [FrameworkElement.Name](#) ,
[FrameworkElementOverridesDefaultStyle](#) , [FrameworkElement.Parent](#) ,
[FrameworkElement.Resources](#) , [FrameworkElement.Style](#) , [FrameworkElement.Tag](#) ,
[FrameworkElement.TemplatedParent](#) , [FrameworkElement.ToolTip](#) , [FrameworkElement.Triggers](#) ,
[FrameworkElement.UseLayoutRounding](#) , [FrameworkElement.VerticalAlignment](#) ,
[FrameworkElement.VisualChildrenCount](#) , [FrameworkElement.Width](#) ,
[FrameworkElement.ContextMenuClosing](#) , [FrameworkElement.ContextMenuOpening](#) ,
[FrameworkElement.DataContextChanged](#) , [FrameworkElement.Initialized](#) ,
[FrameworkElement.Loaded](#) , [FrameworkElement.RequestBringIntoView](#) ,
[FrameworkElement.SizeChanged](#) , [FrameworkElement.SourceUpdated](#) ,
[FrameworkElement.TargetUpdated](#) , [FrameworkElement.ToolTipClosing](#) ,
[FrameworkElement.ToolTipOpening](#) , [FrameworkElement.Unloaded](#) ,
[UIElement.AllowDropProperty](#) , [UIElement.AreAnyTouchesCapturedProperty](#) ,
[UIElement.AreAnyTouchesCapturedWithinProperty](#) , [UIElement.AreAnyTouchesDirectlyOverProperty](#) ,
[UIElement.AreAnyTouchesOverProperty](#) , [UIElement.BitmapEffectInputProperty](#) ,
[UIElement.BitmapEffectProperty](#) , [UIElement.CacheModeProperty](#) , [UIElement.ClipProperty](#) ,
[UIElement.ClipToBoundsProperty](#) , [UIElement.DragEnterEvent](#) , [UIElement.DragLeaveEvent](#) ,
[UIElement.DragOverEvent](#) , [UIElement.DropEvent](#) , [UIElement.EffectProperty](#) ,
[UIElement.FocusableProperty](#) , [UIElement.GiveFeedbackEvent](#) , [UIElement.GotFocusEvent](#) ,
[UIElement.GotKeyboardFocusEvent](#) , [UIElement.GotMouseCaptureEvent](#) ,
[UIElement.GotStylusCaptureEvent](#) , [UIElement.GotTouchCaptureEvent](#) ,
[UIElement.IsEnabledProperty](#) , [UIElement.FocusedProperty](#) , [UIElement.IsHitTestVisibleProperty](#) ,
[UIElement.IsKeyboardFocusedProperty](#) , [UIElement.IsKeyboardFocusWithinProperty](#) ,
[UIElement.IsManipulationEnabledProperty](#) , [UIElement.MouseCapturedProperty](#) ,
[UIElement.IsMouseCaptureWithinProperty](#) , [UIElement.IsMouseDirectlyOverProperty](#) ,
[UIElement.IsMouseOverProperty](#) , [UIElement.StylusCapturedProperty](#) ,
[UIElement.StylusCaptureWithinProperty](#) , [UIElement.IsStylusDirectlyOverProperty](#) ,
[UIElement.IsStylusOverProperty](#) , [UIElement.VisibleProperty](#) , [UIElement.KeyDownEvent](#) ,
[UIElement.KeyUpEvent](#) , [UIElement.LostFocusEvent](#) , [UIElement.LostKeyboardFocusEvent](#) ,
[UIElement.LostMouseCaptureEvent](#) , [UIElement.LostStylusCaptureEvent](#) ,
[UIElement.LostTouchCaptureEvent](#) , [UIElement.ManipulationBoundaryFeedbackEvent](#) ,
[UIElement.ManipulationCompletedEvent](#) , [UIElement.ManipulationDeltaEvent](#) ,
[UIElement.ManipulationInertiaStartingEvent](#) , [UIElement.ManipulationStartedEvent](#) ,
[UIElement.ManipulationStartingEvent](#) , [UIElement.MouseDownEvent](#) , [UIElement.MouseEnterEvent](#) ,
[UIElement.MouseLeaveEvent](#) , [UIElement.MouseLeftButtonDownEvent](#) ,
[UIElement.MouseLeftButtonUpEvent](#) , [UIElement.MouseMoveEvent](#) ,

[UIElement.MouseRightButtonDownEvent](#) , [UIElement.MouseRightButtonUpEvent](#) ,
[UIElement.MouseUpEvent](#) , [UIElement.MouseWheelEvent](#) , [UIElement.OpacityMaskProperty](#) ,
[UIElement.OpacityProperty](#) , [UIElement.PreviewDragEnterEvent](#) ,
[UIElement.PreviewDragLeaveEvent](#) , [UIElement.PreviewDragOverEvent](#) ,
[UIElement.PreviewDropEvent](#) , [UIElement.PreviewGiveFeedbackEvent](#) ,
[UIElement.PreviewGotKeyboardFocusEvent](#) , [UIElement.PreviewKeyDownEvent](#) ,
[UIElement.PreviewKeyUpEvent](#) , [UIElement.PreviewLostKeyboardFocusEvent](#) ,
[UIElement.PreviewMouseDownEvent](#) , [UIElement.PreviewMouseLeftButtonDownEvent](#) ,
[UIElement.PreviewMouseLeftButtonUpEvent](#) , [UIElement.PreviewMouseMoveEvent](#) ,
[UIElement.PreviewMouseRightButtonDownEvent](#) , [UIElement.PreviewMouseRightButtonUpEvent](#) ,
[UIElement.PreviewMouseUpEvent](#) , [UIElement.PreviewMouseWheelEvent](#) ,
[UIElement.PreviewQueryContinueDragEvent](#) , [UIElement.PreviewStylusButtonDownEvent](#) ,
[UIElement.PreviewStylusButtonUpEvent](#) , [UIElement.PreviewStylusDownEvent](#) ,
[UIElement.PreviewStylusInAirMoveEvent](#) , [UIElement.PreviewStylusInRangeEvent](#) ,
[UIElement.PreviewStylusMoveEvent](#) , [UIElement.PreviewStylusOutOfRangeEvent](#) ,
[UIElement.PreviewStylusSystemGestureEvent](#) , [UIElement.PreviewStylusUpEvent](#) ,
[UIElement.PreviewTextInputEvent](#) , [UIElement.PreviewTouchDownEvent](#) ,
[UIElement.PreviewTouchMoveEvent](#) , [UIElement.PreviewTouchUpEvent](#) ,
[UIElement.QueryContinueDragEvent](#) , [UIElement.QueryCursorEvent](#) ,
[UIElement.RenderTransformOriginProperty](#) , [UIElement.RenderTransformProperty](#) ,
[UIElement.SnapsToDevicePixelsProperty](#) , [UIElement.StylusButtonDownEvent](#) ,
[UIElement.StylusButtonUpEvent](#) , [UIElement.StylusDownEvent](#) , [UIElement.StylusEnterEvent](#) ,
[UIElement.StylusInAirMoveEvent](#) , [UIElement.StylusInRangeEvent](#) , [UIElement.StylusLeaveEvent](#) ,
[UIElement.StylusMoveEvent](#) , [UIElement.StylusOutOfRangeEvent](#) ,
[UIElement.StylusSystemGestureEvent](#) , [UIElement.StylusUpEvent](#) , [UIElement.TextInputEvent](#) ,
[UIElement.TouchDownEvent](#) , [UIElement.TouchEnterEvent](#) , [UIElement.TouchLeaveEvent](#) ,
[UIElement.TouchMoveEvent](#) , [UIElement.TouchUpEvent](#) , [UIElement.UidProperty](#) ,
[UIElement.VisibilityProperty](#) , [UIElement.AddHandler\(RoutedEvent, Delegate\)](#) ,
[UIElement.AddHandler\(RoutedEvent, Delegate, bool\)](#) ,
[UIElement.AddToEventRoute\(EventRoute, RoutedEventArgs\)](#) ,
[UIElement.ApplyAnimationClock\(DependencyProperty, AnimationClock\)](#) ,
[UIElement.ApplyAnimationClock\(DependencyProperty, AnimationClock, HandoffBehavior\)](#) ,
[UIElement.Arrange\(Rect\)](#) , [UIElement.BeginAnimation\(DependencyProperty, AnimationTimeline\)](#) ,
[UIElement.BeginAnimation\(DependencyProperty, AnimationTimeline, HandoffBehavior\)](#) ,
[UIElement.CaptureMouse\(\)](#) , [UIElement.CaptureStylus\(\)](#) , [UIElement.CaptureTouch\(TouchDevice\)](#) ,
[UIElement.Focus\(\)](#) , [UIElement.GetAnimationBaseValue\(DependencyProperty\)](#) ,
[UIElement.HitTestCore\(GeometryHitTestParameters\)](#) , [UIElement.HitTestCore\(PointHitTestParameters\)](#) ,
[UIElement.InputHitTest\(Point\)](#) , [UIElement.InvalidateArrange\(\)](#) , [UIElement.InvalidateMeasure\(\)](#) ,
[UIElement.InvalidateVisual\(\)](#) , [UIElement.Measure\(Size\)](#) ,
[UIElement.OnAccessKey\(AccessKeyEventArgs\)](#) , [UIElement.OnChildDesiredSizeChanged\(UIElement\)](#) ,

[UIElement.OnDragEnter\(DragEventArgs\)](#) , [UIElement.OnDragLeave\(DragEventArgs\)](#) ,
[UIElement.OnDragOver\(DragEventArgs\)](#) , [UIElement.OnDrop\(DragEventArgs\)](#) ,
[UIElement.OnGiveFeedback\(GiveFeedbackEventArgs\)](#) ,
[UIElement.OnGotKeyboardFocus\(KeyboardFocusChangedEventArgs\)](#) ,
[UIElement.OnGotMouseCapture\(MouseEventArgs\)](#) ,
[UIElement.OnGotStylusCapture\(StylusEventArgs\)](#) , [UIElement.OnGotTouchCapture\(TouchEventArgs\)](#) ,
[UIElement.OnIsKeyboardFocusedChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsKeyboardFocusWithinChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsMouseCapturedChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsMouseCaptureWithinChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsMouseDirectlyOverChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsStylusCapturedChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsStylusCaptureWithinChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsStylusDirectlyOverChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnKeyDown\(KeyEventArgs\)](#) , [UIElement.OnKeyUp\(KeyEventArgs\)](#) ,
[UIElement.OnLostFocus\(RoutedEventArgs\)](#) ,
[UIElement.OnLostKeyboardFocus\(KeyboardFocusChangedEventArgs\)](#) ,
[UIElement.OnLostMouseCapture\(MouseEventArgs\)](#) ,
[UIElement.OnLostStylusCapture\(StylusEventArgs\)](#) , [UIElement.OnLostTouchCapture\(TouchEventArgs\)](#) ,
[UIElement.OnManipulationCompleted\(ManipulationCompletedEventArgs\)](#) ,
[UIElement.OnManipulationDelta\(ManipulationDeltaEventArgs\)](#) ,
[UIElement.OnManipulationInertiaStarting\(ManipulationInertiaStartingEventArgs\)](#) ,
[UIElement.OnManipulationStarted\(ManipulationStartedEventArgs\)](#) ,
[UIElement.OnManipulationStarting\(ManipulationStartingEventArgs\)](#) ,
[UIElement.OnMouseDown\(MouseButtonEventArgs\)](#) , [UIElement.OnMouseEnter\(MouseEventArgs\)](#) ,
[UIElement.OnMouseLeave\(MouseEventArgs\)](#) ,
[UIElement.OnMouseLeftButtonDown\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseLeftButtonUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseMove\(MouseEventArgs\)](#) ,
[UIElement.OnMouseRightButtonDown\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseRightButtonUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseWheel\(MouseEventArgs\)](#) ,
[UIElement.OnPreviewDragEnter\(DragEventArgs\)](#) , [UIElement.OnPreviewDragLeave\(DragEventArgs\)](#) ,
[UIElement.OnPreviewDragOver\(DragEventArgs\)](#) , [UIElement.OnPreviewDrop\(DragEventArgs\)](#) ,
[UIElement.OnPreviewGiveFeedback\(GiveFeedbackEventArgs\)](#) ,
[UIElement.OnPreviewGotKeyboardFocus\(KeyboardFocusChangedEventArgs\)](#) ,
[UIElement.OnPreviewKeyDown\(KeyEventArgs\)](#) , [UIElement.OnPreviewKeyUp\(KeyEventArgs\)](#) ,
[UIElement.OnPreviewLostKeyboardFocus\(KeyboardFocusChangedEventArgs\)](#) ,
[UIElement.OnPreviewMouseDown\(MouseButtonEventArgs\)](#) ,

[UIElement.OnPreviewMouseLeftButtonDown\(MouseEventArgs\)](#) ,
[UIElement.OnPreviewMouseLeftButtonUp\(MouseEventArgs\)](#) ,
[UIElement.OnPreviewMouseMove\(MouseEventArgs\)](#) ,
[UIElement.OnPreviewMouseRightButtonDown\(MouseEventArgs\)](#) ,
[UIElement.OnPreviewMouseRightButtonUp\(MouseEventArgs\)](#) ,
[UIElement.OnPreviewMouseUp\(MouseEventArgs\)](#) ,
[UIElement.OnPreviewMouseWheel\(MouseWheelEventArgs\)](#) ,
[UIElement.OnPreviewQueryContinueDrag\(QueryContinueDragEventArgs\)](#) ,
[UIElement.OnPreviewStylusButtonDown\(StylusButtonEventArgs\)](#) ,
[UIElement.OnPreviewStylusButtonUp\(StylusButtonEventArgs\)](#) ,
[UIElement.OnPreviewStylusDown\(StylusDownEventArgs\)](#) ,
[UIElement.OnPreviewStylusInAirMove\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewStylusInRange\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewStylusMove\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewStylusOutOfRange\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewStylusSystemGesture\(StylusSystemGestureEventArgs\)](#) ,
[UIElement.OnPreviewStylusUp\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewTextInput\(TextCompositionEventArgs\)](#) ,
[UIElement.OnPreviewTouchDown\(TouchEventArgs\)](#) ,
[UIElement.OnPreviewTouchMove\(TouchEventArgs\)](#) , [UIElement.OnPreviewTouchUp\(TouchEventArgs\)](#) ,
[UIElement.OnQueryContinueDrag\(QueryContinueDragEventArgs\)](#) ,
[UIElement.OnQueryCursor\(QueryCursorEventArgs\)](#) , [UIElement.OnRender\(DrawingContext\)](#) ,
[UIElement.OnStylusButtonDown\(StylusButtonEventArgs\)](#) ,
[UIElement.OnStylusButtonUp\(StylusButtonEventArgs\)](#) ,
[UIElement.OnStylusDown\(StylusDownEventArgs\)](#) , [UIElement.OnStylusEnter\(StylusEventArgs\)](#) ,
[UIElement.OnStylusInAirMove\(StylusEventArgs\)](#) , [UIElement.OnStylusInRange\(StylusEventArgs\)](#) ,
[UIElement.OnStylusLeave\(StylusEventArgs\)](#) , [UIElement.OnStylusMove\(StylusEventArgs\)](#) ,
[UIElement.OnStylusOutOfRange\(StylusEventArgs\)](#) ,
[UIElement.OnStylusSystemGesture\(StylusSystemGestureEventArgs\)](#) ,
[UIElement.OnStylusUp\(StylusEventArgs\)](#) , [UIElement.OnTextInput\(TextCompositionEventArgs\)](#) ,
[UIElement.OnTouchDown\(TouchEventArgs\)](#) , [UIElement.OnTouchEnter\(TouchEventArgs\)](#) ,
[UIElement.OnTouchLeave\(TouchEventArgs\)](#) , [UIElement.OnTouchMove\(TouchEventArgs\)](#) ,
[UIElement.OnTouchUp\(TouchEventArgs\)](#) , [UIElement.RaiseEvent\(RoutedEventArgs\)](#) ,
[UIElement.ReleaseAllTouchCaptures\(\)](#) , [UIElement.ReleaseMouseCapture\(\)](#) ,
[UIElement.ReleaseStylusCapture\(\)](#) , [UIElement.ReleaseTouchCapture\(TouchDevice\)](#) ,
[UIElement.RemoveHandler\(RoutedEvent, Delegate\)](#) , [UIElement.TranslatePoint\(Point, UIElement\)](#) ,
[UIElement.UpdateLayout\(\)](#) , [UIElement.AllowDrop](#) , [UIElement.AreAnyTouchesCaptured](#) ,
[UIElement.AreAnyTouchesCapturedWithin](#) , [UIElement.AreAnyTouchesDirectlyOver](#) ,
[UIElement.AreAnyTouchesOver](#) , [UIElement.BitmapEffect](#) , [UIElement.BitmapEffectInput](#) ,
[UIElement.CacheMode](#) , [UIElement.Clip](#) , [UIElement.ClipToBounds](#) ,

[UIElement.CommandBindings](#) , [UIElement.DesiredSize](#) , [UIElement.Effect](#) , [UIElement.Focusable](#) ,
[UIElement.HasAnimatedProperties](#) , [UIElement.HasEffectiveKeyboardFocus](#) ,
[UIElement.InputBindings](#) , [UIElement.IsArrangeValid](#) , [UIElement.IsEnabled](#) ,
[UIElement.IsEnabledCore](#) , [UIElement.IsFocused](#) , [UIElement.IsHitTestVisible](#) ,
[UIElement.IsInputMethodEnabled](#) , [UIElement.IsKeyboardFocused](#) ,
[UIElement.IsKeyboardFocusWithin](#) , [UIElement.IsManipulationEnabled](#) , [UIElement.IsMeasureValid](#) ,
[UIElement.IsMouseCaptured](#) , [UIElement.IsMouseCaptureWithin](#) , [UIElement.IsMouseDirectlyOver](#) ,
[UIElement.IsMouseOver](#) , [UIElement.IsStylusCaptured](#) , [UIElement.IsStylusCaptureWithin](#) ,
[UIElement.IsStylusDirectlyOver](#) , [UIElement.IsStylusOver](#) , [UIElement.IsVisible](#) , [UIElement.Opacity](#) ,
[UIElement.OpacityMask](#) , [UIElement.PersistId](#) , [UIElement.RenderSize](#) ,
[UIElement.RenderTransform](#) , [UIElement.RenderTransformOrigin](#) , [UIElement.SnapsToDevicePixels](#) ,
[UIElement.StylusPlugIns](#) , [UIElement.TouchesCaptured](#) , [UIElement.TouchesCapturedWithin](#) ,
[UIElement.TouchesDirectlyOver](#) , [UIElement.TouchesOver](#) , [UIElement.Uid](#) , [UIElement.Visibility](#) ,
[UIElement.DragEnter](#) , [UIElement.DragLeave](#) , [UIElement.DragOver](#) , [UIElement.Drop](#) ,
[UIElement.FocusableChanged](#) , [UIElement.GiveFeedback](#) , [UIElement.GotFocus](#) ,
[UIElement.GotKeyboardFocus](#) , [UIElement.GotMouseCapture](#) , [UIElement.GotStylusCapture](#) ,
[UIElement.GotTouchCapture](#) , [UIElement.IsEnabledChanged](#) , [UIElement.IsHitTestVisibleChanged](#) ,
[UIElement.IsKeyboardFocusedChanged](#) , [UIElement.IsKeyboardFocusWithinChanged](#) ,
[UIElement.IsMouseCapturedChanged](#) , [UIElement.IsMouseCaptureWithinChanged](#) ,
[UIElement.IsMouseDirectlyOverChanged](#) , [UIElement.IsStylusCapturedChanged](#) ,
[UIElement.IsStylusCaptureWithinChanged](#) , [UIElement.IsStylusDirectlyOverChanged](#) ,
[UIElement.IsVisibleChanged](#) , [UIElement.KeyDown](#) , [UIElement.KeyUp](#) ,
[UIElement.LayoutUpdated](#) , [UIElement.LostFocus](#) , [UIElement.LostKeyboardFocus](#) ,
[UIElement.LostMouseCapture](#) , [UIElement.LostStylusCapture](#) , [UIElement.LostTouchCapture](#) ,
[UIElement.ManipulationBoundaryFeedback](#) , [UIElement.ManipulationCompleted](#) ,
[UIElement.ManipulationDelta](#) , [UIElement.ManipulationInertiaStarting](#) ,
[UIElement.ManipulationStarted](#) , [UIElement.ManipulationStarting](#) , [UIElement.MouseDown](#) ,
[UIElement.MouseEnter](#) , [UIElement.MouseLeave](#) , [UIElement.MouseLeftButtonDown](#) ,
[UIElement.MouseLeftButtonUp](#) , [UIElement.MouseMove](#) , [UIElement.MouseRightButtonDown](#) ,
[UIElement.MouseRightButtonUp](#) , [UIElement.MouseUp](#) , [UIElement.MouseWheel](#) ,
[UIElement.PreviewDragEnter](#) , [UIElement.PreviewDragLeave](#) , [UIElement.PreviewDragOver](#) ,
[UIElement.PreviewDrop](#) , [UIElement.PreviewGiveFeedback](#) , [UIElement.PreviewGotKeyboardFocus](#) ,
[UIElement.PreviewKeyDown](#) , [UIElement.PreviewKeyUp](#) , [UIElement.PreviewLostKeyboardFocus](#) ,
[UIElement.PreviewMouseDown](#) , [UIElement.PreviewMouseLeftButtonDown](#) ,
[UIElement.PreviewMouseLeftButtonUp](#) , [UIElement.PreviewMouseMove](#) ,
[UIElement.PreviewMouseRightButtonDown](#) , [UIElement.PreviewMouseRightButtonUp](#) ,
[UIElement.PreviewMouseUp](#) , [UIElement.PreviewMouseWheel](#) ,
[UIElement.PreviewQueryContinueDrag](#) , [UIElement.PreviewStylusButtonDown](#) ,
[UIElement.PreviewStylusButtonUp](#) , [UIElement.PreviewStylusDown](#) ,
[UIElement.PreviewStylusInAirMove](#) , [UIElement.PreviewStylusInRange](#) ,

[UIElement.PreviewStylusMove](#) , [UIElement.PreviewStylusOutOfRange](#) ,
[UIElement.PreviewStylusSystemGesture](#) , [UIElement.PreviewStylusUp](#) , [UIElement.PreviewTextInput](#) ,
[UIElement.PreviewTouchDown](#) , [UIElement.PreviewTouchMove](#) , [UIElement.PreviewTouchUp](#) ,
[UIElement.QueryContinueDrag](#) , [UIElement.QueryCursor](#) , [UIElement.StylusButtonDown](#) ,
[UIElement.StylusButtonUp](#) , [UIElement.StylusDown](#) , [UIElement.StylusEnter](#) ,
[UIElement.StylusInAirMove](#) , [UIElement.StylusInRange](#) , [UIElement.StylusLeave](#) ,
[UIElement.StylusMove](#) , [UIElement.StylusOutOfRange](#) , [UIElement.StylusSystemGesture](#) ,
[UIElement.StylusUp](#) , [UIElement.TextInput](#) , [UIElement.TouchDown](#) , [UIElement.TouchEnter](#) ,
[UIElement.TouchLeave](#) , [UIElement.TouchMove](#) , [UIElement.TouchUp](#) ,
[Visual.AddVisualChild\(Visual\)](#) , [Visual.FindCommonVisualAncestor\(DependencyObject\)](#) ,
[Visual.IsAncestorOf\(DependencyObject\)](#) , [Visual.IsDescendantOf\(DependencyObject\)](#) ,
[Visual.PointFromScreen\(Point\)](#) , [Visual.PointToScreen\(Point\)](#) , [Visual.RemoveVisualChild\(Visual\)](#) ,
[Visual.TransformToAncestor\(Visual3D\)](#) , [Visual.TransformToAncestor\(Visual\)](#) ,
[Visual.TransformToDescendant\(Visual\)](#) , [Visual.TransformToVisual\(Visual\)](#) , [Visual.VisualBitmapEffect](#) ,
[Visual.VisualBitmapEffectInput](#) , [Visual.VisualBitmapScalingMode](#) , [Visual.VisualCacheMode](#) ,
[Visual.VisualClearTypeHint](#) , [Visual.VisualClip](#) , [Visual.VisualEdgeMode](#) , [Visual.VisualEffect](#) ,
[Visual.VisualOffset](#) , [Visual.VisualOpacity](#) , [Visual.VisualOpacityMask](#) , [Visual.VisualParent](#) ,
[Visual.VisualScrollableAreaClip](#) , [Visual.VisualTextHintingMode](#) , [Visual.VisualTextRenderingMode](#) ,
[Visual.VisualTransform](#) , [Visual.VisualXSnappingGuidelines](#) , [Visual.VisualYSnappingGuidelines](#) ,
[DependencyObject.ClearValue\(DependencyProperty\)](#) ,
[DependencyObject.ClearValue\(DependencyPropertyKey\)](#) ,
[DependencyObject.CoerceValue\(DependencyProperty\)](#) , [DependencyObject.Equals\(object\)](#) ,
[DependencyObject.GetHashCode\(\)](#) , [DependencyObject.GetLocalValueEnumerator\(\)](#) ,
[DependencyObject.GetValue\(DependencyProperty\)](#) ,
[DependencyObject.InvalidateProperty\(DependencyProperty\)](#) ,
[DependencyObject.ReadLocalValue\(DependencyProperty\)](#) ,
[DependencyObject.SetCurrentValue\(DependencyProperty, object\)](#) ,
[DependencyObject.SetValue\(DependencyProperty, object\)](#) ,
[DependencyObject.SetValue\(DependencyPropertyKey, object\)](#) ,
[DependencyObject.ShouldSerializeProperty\(DependencyProperty\)](#) ,
[DependencyObject.DependencyObjectType](#) , [DependencyObject.IsSealed](#) ,
[DispatcherObject.Dispatcher](#) , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

CanvasTopic(string, ref Canvas)

Draws the nets structure onto a given 'Canvas'.

Additionally you can open a 'CanvasWindow' with the diagramm.

```
public CanvasTopic(string windowTitle, ref Canvas inCanvas)
```

Parameters

windowTitle [string](#)

title of the window

inCanvas [Canvas](#)

the to used 'Canvas'

Fields

basicTopic

Basic topicstring - minimum for an FFN.

```
public string basicTopic
```

Field Value

[string](#)

topicField

The layers as int[] - 'parsed' from the topic string.

```
public int[] topicField
```

Field Value

[int](#)[]

version

created on: 08.07.2023

last edit: 20.09.24

```
public Version version
```

Field Value

[Version](#) ↗

workingTopic

Topic in the course of the program.

```
public string workingTopic
```

Field Value

[string](#) ↗

Methods

ParseDataIntoTopic(double[][], double[][])

The given dataset defines the input-output layers.

```
public bool ParseDataIntoTopic(double[][] datenIn, double[][] datenOut)
```

Parameters

datenIn [double](#)[][]

datenOut [double](#)[][]

Returns

[bool](#) ↗

ParseTopic(string, ref int[])

Parses the int[] of the layers from a string representation.

```
public bool ParseTopic(string topic, ref int[] layerInts)
```

Parameters

topic [string](#)

the topic string

layerInts [int](#)[]

referene to a int[] for the layers

Returns

[bool](#)

success of the operation

PutCanvasWindowEllipse(double, double, double, double)

Helperfunction putting an ellipse into 'canvasWindow' at the given coordinates.

```
public void PutCanvasWindowEllipse(double xPos, double yPos, double height = 20, double width = 20)
```

Parameters

xPos [double](#)

distance from the left border minus half of the width

yPos [double](#)

distance from th top border minus half the height

height [double](#)

standard if not changed

width [double](#)

standard if not changed

PutCanvasWindowLabel(double, double, string, double, double)

Helperfunction for 'canvasWindow' putting a label onto the given coordinates.

```
public void PutCanvasWindowLabel(double xPos, double yPos, string text, double height = 20,  
double width = 20)
```

Parameters

xPos [double](#)

distance from the left border

yPos [double](#)

distance from the top border

text [string](#)

the message

height [double](#)

standard if not changed

width [double](#)

standard if not changed

PutCanvasWindowLine(double, double, double, double, int, Brush)

Helperfunction settin a line into 'canvasWindow'.

```
public void PutCanvasWindowLine(double x1, double y1, double x2, double y2, int thickness,  
Brush colBrush)
```

Parameters

x1 [double](#)

from x-coordinte

y1 [double](#)

from y-coordinate

x2 [double](#)

to x-coordinate

y2 [double](#)

to y-coordinate

thickness [int](#)

standard broadness is 2

colBrush [Brush](#)

color of the brush

PutCanvasWindowLineStroked(double, double, double, double, int, Brush)

Helperfunction settin a stroked line into 'canvasWindow'.

```
public void PutCanvasWindowLineStroked(double x1, double y1, double x2, double y2, int thickness, Brush colBrush)
```

Parameters

x1 [double](#)

from x-coordinte

y1 [double](#)

from y-coordinate

x2 [double](#)

to x-coordinate

y2 [double](#)

to y-coordinate

thickness [int](#)

standard broadness is 2

colBrush [Brush](#)

color of the brush

PutDrawAreaEllipse(double, double, double, double)

Helperfunction putting an ellipse into the 'drawArea'.

```
public void PutDrawAreaEllipse(double xPos, double yPos, double height = 20, double width = 20)
```

Parameters

xPos [double](#)

distance from the left border minus the half width

yPos [double](#)

distance from the top border minus the half height

height [double](#)

standard if not changed

width [double](#)

standard if not changed

PutDrawAreaLabel(double, double, string, double, double)

Helperfunction putting a label onto the given coordinates in the 'drawArea'.

```
public void PutDrawAreaLabel(double xPos, double yPos, string text, double height = 20,  
double width = 20)
```

Parameters

xPos [double](#)

distance from the left border

yPos [double](#)

distance from the top border

text [string](#)

message

height [double](#)

standard if not changed

width [double](#)

standard if not changed

PutDrawAreaLine(double, double, double, double, int, Brush)

Helperfunction putting a line into the 'drawArea'.

```
public void PutDrawAreaLine(double x1, double y1, double x2, double y2, int thickness,  
Brush colBrush)
```

Parameters

x1 [double](#)

from x-coordinate

y1 [double](#)

from y-coordinate

x2 [double](#)

to x-coordinate

y2 [double](#)

to y-coordinate

thickness [int](#)

standard broadness is 2

colBrush [Brush](#)

standard color is Brushes.Black

PutDrawAreaLineStroked(double, double, double, double, int, Brush)

Helperfunction putting a stroked line into the 'drawArea'.

```
public void PutDrawAreaLineStroked(double x1, double y1, double x2, double y2, int thickness, Brush colBrush)
```

Parameters

x1 [double](#)

from x-coordinate

y1 [double](#)

from y-coordinate

x2 [double](#)

to x-coordinate

y2 [double](#)

to y-coordinate

thickness [int ↗](#)

standard broadness is 2

colBrush [Brush ↗](#)

standard color is Brushes.Black

ShowTopic()

Shows the nets diagram in the 'drawArea'.

```
public void ShowTopic()
```

ShowWindow()

Shows the nets graph in the 'canvasWindow' - a zoomed window.

```
public void ShowWindow()
```

Class CanvasWindow

Namespace: [MatrixFFN.Tools](#)

Assembly: MatrixFFN.dll

Interaktionslogik für CanvasFenster.xaml

```
public class CanvasWindow : Window, IAnimatable, ISupportInitialize, IFrameworkInputElement,  
IInputElement, IQueryAmbient, IAddChild, IComponentConnector
```

Inheritance

```
object ← DispatcherObject ← DependencyObject ← Visual ← UIElement ←  
FrameworkElement ← Control ← ContentControl ← Window ← CanvasWindow
```

Implements

```
IAnimatable, ISupportInitialize, IFrameworkInputElement, IInputElement, IQueryAmbient,  
IAddChild, IComponentConnector
```

Derived

[CanvasChart](#), [CanvasTopic](#)

Inherited Members

```
Window.AllowsTransparencyProperty, Window.DpiChangedEvent, Window.IconProperty,  
Window.IsActiveProperty, Window.LeftProperty, Window.ResizeModeProperty,  
Window.ShowActivatedProperty, Window.ShowInTaskbarProperty,  
Window.SizeToContentProperty, Window.TaskBarItemInfoProperty, Window.TitleProperty,  
Window.TopmostProperty, Window.TopProperty, Window.WindowStateProperty,  
Window.WindowStyleProperty, Window.Activate(), Window.ArrangeOverride(Size),  
Window.Close(), Window.DragMove(), Window.GetWindow(DependencyObject),  
Window.Hide(), Window.MeasureOverride(Size), Window.OnActivated(EventArgs),  
Window.OnClosed(EventArgs), Window.OnClosing(CancelEventArgs),  
Window.OnContentChanged(object, object), Window.OnContentRendered(EventArgs),  
Window.OnCreateAutomationPeer(), Window.OnDeactivated(EventArgs),  
Window.OnDpiChanged(DpiScale, DpiScale), Window.OnLocationChanged(EventArgs),  
Window.OnManipulationBoundaryFeedback(ManipulationBoundaryFeedbackEventArgs),  
Window.OnSourceInitialized(EventArgs), Window.OnStateChanged(EventArgs),  
Window.OnVisualChildrenChanged(DependencyObject, DependencyObject),  
Window.OnVisualParentChanged(DependencyObject), Window.Show(), Window.ShowDialog(),  
WindowAllowsTransparency, Window.DialogResult, Window.Icon, Window.IsActive,  
Window.Left, WindowLogicalChildren, WindowOwnedWindows, WindowOwner,
```

[Window.ResizeMode](#) , [Window.RestoreBounds](#) , [Window>ShowActivated](#) ,
[Window>ShowInTaskbar](#) , [Window>SizeToContent](#) , [Window>TaskBarItemInfo](#) , [Window>Title](#) ,
[Window>Top](#) , [Window>Topmost](#) , [Window>WindowStartupLocation](#) , [Window>WindowState](#) ,
[Window>WindowStyle](#) , [Window>Activated](#) , [Window>Closed](#) , [Window>Closing](#) ,
[Window>ContentRendered](#) , [Window>Deactivated](#) , [Window>DpiChanged](#) ,
[Window>LocationChanged](#) , [Window>SourceInitialized](#) , [Window>StateChanged](#) ,
[ContentControl>ContentProperty](#) , [ContentControl>ContentStringFormatProperty](#) ,
[ContentControl>ContentTemplateProperty](#) , [ContentControl>ContentTemplateSelectorProperty](#) ,
[ContentControl>HasContentProperty](#) , [ContentControl>AddChild\(object\)](#) ,
[ContentControl>AddText\(string\)](#) , [ContentControl>OnContentStringFormatChanged\(string, string\)](#) ,
[ContentControl>OnContentTemplateChanged\(DataTemplate, DataTemplate\)](#) ,
[ContentControl>OnContentTemplateSelectorChanged\(DataTemplateSelector, DataTemplateSelector\)](#) ,
[ContentControl>Content](#) , [ContentControl>ContentStringFormat](#) , [ContentControl>ContentTemplate](#) ,
[ContentControl>ContentTemplateSelector](#) , [ContentControl>HasContent](#) ,
[Control>BackgroundProperty](#) , [Control>BorderBrushProperty](#) , [Control>BorderThicknessProperty](#) ,
[Control>FontFamilyProperty](#) , [Control>FontSizeProperty](#) , [Control>FontStretchProperty](#) ,
[Control>FontStyleProperty](#) , [Control>FontWeightProperty](#) , [Control>ForegroundProperty](#) ,
[Control>HorizontalContentAlignmentProperty](#) , [Control>IsTabStopProperty](#) ,
[Control>MouseDoubleClickEvent](#) , [Control>PaddingProperty](#) ,
[Control>PreviewMouseDoubleClickEvent](#) , [Control>TabIndexProperty](#) , [Control>TemplateProperty](#) ,
[Control>VerticalContentAlignmentProperty](#) , [Control>OnMouseDoubleClick\(MouseEventArgs\)](#) ,
[Control>OnPreviewMouseDoubleClick\(MouseEventArgs\)](#) ,
[Control>OnTemplateChanged\(ControlTemplate, ControlTemplate\)](#) , [Control>ToString\(\)](#) ,
[Control>Background](#) , [Control>BorderBrush](#) , [Control>BorderThickness](#) , [Control>FontFamily](#) ,
[Control>FontSize](#) , [Control>FontStretch](#) , [Control>FontStyle](#) , [Control>FontWeight](#) ,
[Control>Foreground](#) , [Control>HandlesScrolling](#) , [Control>HorizontalContentAlignment](#) ,
[Control>IsTabStop](#) , [Control>Padding](#) , [Control>TabIndex](#) , [Control>Template](#) ,
[Control>VerticalContentAlignment](#) , [Control>MouseDoubleClick](#) ,
[Control>PreviewMouseDoubleClick](#) , [FrameworkElement>ActualHeightProperty](#) ,
[FrameworkElement>ActualWidthProperty](#) , [FrameworkElement>BindingGroupProperty](#) ,
[FrameworkElement>ContextMenuClosingEvent](#) , [FrameworkElement>ContextMenuOpeningEvent](#) ,
[FrameworkElement>ContextMenuProperty](#) , [FrameworkElement>CursorProperty](#) ,
[FrameworkElement>DataContextProperty](#) , [FrameworkElement>DefaultStyleKeyProperty](#) ,
[FrameworkElement>FlowDirectionProperty](#) , [FrameworkElement>FocusVisualStyleProperty](#) ,
[FrameworkElement>ForceCursorProperty](#) , [FrameworkElement>HeightProperty](#) ,
[FrameworkElement>HorizontalAlignmentProperty](#) , [FrameworkElement>InputScopeProperty](#) ,
[FrameworkElement>LanguageProperty](#) , [FrameworkElement>LayoutTransformProperty](#) ,
[FrameworkElement>LoadedEvent](#) , [FrameworkElement>MarginProperty](#) ,
[FrameworkElement>MaxHeightProperty](#) , [FrameworkElement>MaxWidthProperty](#) ,
[FrameworkElement>MinHeightProperty](#) , [FrameworkElement>MinWidthProperty](#) ,

[FrameworkElement.NameProperty](#) , [FrameworkElement.OverridesDefaultStyleProperty](#) ,
[FrameworkElement.RequestBringIntoViewEvent](#) , [FrameworkElement.SizeChangedEvent](#) ,
[FrameworkElement.StyleProperty](#) , [FrameworkElement.TagProperty](#) ,
[FrameworkElement.ToolTipClosingEvent](#) , [FrameworkElement.ToolTipOpeningEvent](#) ,
[FrameworkElement.ToolTipProperty](#) , [FrameworkElement.UnloadedEvent](#) ,
[FrameworkElement.UseLayoutRoundingProperty](#) , [FrameworkElement.VerticalAlignmentProperty](#) ,
[FrameworkElement.WidthProperty](#) , [FrameworkElement.AddLogicalChild\(object\)](#) ,
[FrameworkElement.ApplyTemplate\(\)](#) , [FrameworkElement.ArrangeCore\(Rect\)](#) ,
[FrameworkElement.BeginInit\(\)](#) , [FrameworkElement.BeginStoryboard\(Storyboard\)](#) ,
[FrameworkElement.BeginStoryboard\(Storyboard, HandoffBehavior\)](#) ,
[FrameworkElement.BeginStoryboard\(Storyboard, HandoffBehavior, bool\)](#) ,
[FrameworkElement.BringIntoView\(\)](#) , [FrameworkElement.BringIntoView\(Rect\)](#) ,
[FrameworkElement.EndInit\(\)](#) , [FrameworkElement.FindName\(string\)](#) ,
[FrameworkElement.FindResource\(object\)](#) ,
[FrameworkElement.GetBindingExpression\(DependencyProperty\)](#) ,
[FrameworkElement.GetFlowDirection\(DependencyObject\)](#) , [FrameworkElement.GetLayoutClip\(Size\)](#) ,
[FrameworkElement.GetTemplateChild\(string\)](#) , [FrameworkElement.GetUIParentCore\(\)](#) ,
[FrameworkElement.GetVisualChild\(int\)](#) , [FrameworkElement.MeasureCore\(Size\)](#) ,
[FrameworkElement.MoveFocus\(TraversalRequest\)](#) , [FrameworkElement.OnApplyTemplate\(\)](#) ,
[FrameworkElement.OnContextMenuClosing\(ContextMenuEventArgs\)](#) ,
[FrameworkElement.OnContextMenuOpening\(ContextMenuEventArgs\)](#) ,
[FrameworkElement.OnGotFocus\(RoutedEventArgs\)](#) , [FrameworkElement.OnInitialized\(EventArgs\)](#) ,
[FrameworkElement.OnPropertyChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[FrameworkElement.OnRenderSizeChanged\(SizeChangedEventArgs\)](#) ,
[FrameworkElement.OnStyleChanged\(Style, Style\)](#) ,
[FrameworkElement.OnToolTipClosing\(ToolTipEventArgs\)](#) ,
[FrameworkElement.OnToolTipOpening\(ToolTipEventArgs\)](#) ,
[FrameworkElement.ParentLayoutInvalidated\(UIElement\)](#) ,
[FrameworkElement.PredictFocus\(FocusNavigationDirection\)](#) ,
[FrameworkElement.RegisterName\(string, object\)](#) , [FrameworkElement.RemoveLogicalChild\(object\)](#) ,
[FrameworkElement.SetBinding\(DependencyProperty, string\)](#) ,
[FrameworkElement.SetBinding\(DependencyProperty, BindingBase\)](#) ,
[FrameworkElement.SetFlowDirection\(DependencyObject, FlowDirection\)](#) ,
[FrameworkElement.SetResourceReference\(DependencyProperty, object\)](#) ,
[FrameworkElement.TryFindResource\(object\)](#) , [FrameworkElement.UnregisterName\(string\)](#) ,
[FrameworkElement.UpdateDefaultStyle\(\)](#) , [FrameworkElement.ActualHeight](#) ,
[FrameworkElement.ActualWidth](#) , [FrameworkElement.BindingGroup](#) ,
[FrameworkElement.ContextMenu](#) , [FrameworkElement.Cursor](#) , [FrameworkElement.DataContext](#) ,
[FrameworkElement.DefaultStyleKey](#) , [FrameworkElement.FlowDirection](#) ,
[FrameworkElement.FocusVisualStyle](#) , [FrameworkElement.ForceCursor](#) , [FrameworkElement.Height](#) ,

[FrameworkElement.HorizontalAlignment](#) , [FrameworkElement.InheritanceBehavior](#) ,
[FrameworkElement.InputScope](#) , [FrameworkElement.IsInitialized](#) , [FrameworkElement.IsLoaded](#) ,
[FrameworkElement.Language](#) , [FrameworkElement.LayoutTransform](#) , [FrameworkElement.Margin](#) ,
[FrameworkElement.MaxHeight](#) , [FrameworkElement.MaxWidth](#) , [FrameworkElement.MinHeight](#) ,
[FrameworkElement.MinWidth](#) , [FrameworkElement.Name](#) ,
[FrameworkElementOverridesDefaultStyle](#) , [FrameworkElement.Parent](#) ,
[FrameworkElement.Resources](#) , [FrameworkElement.Style](#) , [FrameworkElement.Tag](#) ,
[FrameworkElement.TemplatedParent](#) , [FrameworkElement.ToolTip](#) , [FrameworkElement.Triggers](#) ,
[FrameworkElement.UseLayoutRounding](#) , [FrameworkElement.VerticalAlignment](#) ,
[FrameworkElement.VisualChildrenCount](#) , [FrameworkElement.Width](#) ,
[FrameworkElement.ContextMenuClosing](#) , [FrameworkElement.ContextMenuOpening](#) ,
[FrameworkElement.DataContextChanged](#) , [FrameworkElement.Initialized](#) ,
[FrameworkElement.Loaded](#) , [FrameworkElement.RequestBringIntoView](#) ,
[FrameworkElement.SizeChanged](#) , [FrameworkElement.SourceUpdated](#) ,
[FrameworkElement.TargetUpdated](#) , [FrameworkElement.ToolTipClosing](#) ,
[FrameworkElement.ToolTipOpening](#) , [FrameworkElement.Unloaded](#) ,
[UIElement.AllowDropProperty](#) , [UIElement.AreAnyTouchesCapturedProperty](#) ,
[UIElement.AreAnyTouchesCapturedWithinProperty](#) , [UIElement.AreAnyTouchesDirectlyOverProperty](#) ,
[UIElement.AreAnyTouchesOverProperty](#) , [UIElement.BitmapEffectInputProperty](#) ,
[UIElement.BitmapEffectProperty](#) , [UIElement.CacheModeProperty](#) , [UIElement.ClipProperty](#) ,
[UIElement.ClipToBoundsProperty](#) , [UIElement.DragEnterEvent](#) , [UIElement.DragLeaveEvent](#) ,
[UIElement.DragOverEvent](#) , [UIElement.DropEvent](#) , [UIElement.EffectProperty](#) ,
[UIElement.FocusableProperty](#) , [UIElement.GiveFeedbackEvent](#) , [UIElement.GotFocusEvent](#) ,
[UIElement.GotKeyboardFocusEvent](#) , [UIElement.GotMouseCaptureEvent](#) ,
[UIElement.GotStylusCaptureEvent](#) , [UIElement.GotTouchCaptureEvent](#) ,
[UIElement.IsEnabledProperty](#) , [UIElement.FocusedProperty](#) , [UIElement.IsHitTestVisibleProperty](#) ,
[UIElement.IsKeyboardFocusedProperty](#) , [UIElement.IsKeyboardFocusWithinProperty](#) ,
[UIElement.IsManipulationEnabledProperty](#) , [UIElement.MouseCapturedProperty](#) ,
[UIElement.IsMouseCaptureWithinProperty](#) , [UIElement.IsMouseDirectlyOverProperty](#) ,
[UIElement.IsMouseOverProperty](#) , [UIElement.StylusCapturedProperty](#) ,
[UIElement.StylusCaptureWithinProperty](#) , [UIElement.IsStylusDirectlyOverProperty](#) ,
[UIElement.IsStylusOverProperty](#) , [UIElement.VisibleProperty](#) , [UIElement.KeyDownEvent](#) ,
[UIElement.KeyUpEvent](#) , [UIElement.LostFocusEvent](#) , [UIElement.LostKeyboardFocusEvent](#) ,
[UIElement.LostMouseCaptureEvent](#) , [UIElement.LostStylusCaptureEvent](#) ,
[UIElement.LostTouchCaptureEvent](#) , [UIElement.ManipulationBoundaryFeedbackEvent](#) ,
[UIElement.ManipulationCompletedEvent](#) , [UIElement.ManipulationDeltaEvent](#) ,
[UIElement.ManipulationInertiaStartingEvent](#) , [UIElement.ManipulationStartedEvent](#) ,
[UIElement.ManipulationStartingEvent](#) , [UIElement.MouseDownEvent](#) , [UIElement.MouseEnterEvent](#) ,
[UIElement.MouseLeaveEvent](#) , [UIElement.MouseLeftButtonDownEvent](#) ,
[UIElement.MouseLeftButtonUpEvent](#) , [UIElement.MouseMoveEvent](#) ,

[UIElement.MouseRightButtonDownEvent](#) , [UIElement.MouseRightButtonUpEvent](#) ,
[UIElement.MouseUpEvent](#) , [UIElement.MouseWheelEvent](#) , [UIElement.OpacityMaskProperty](#) ,
[UIElement.OpacityProperty](#) , [UIElement.PreviewDragEnterEvent](#) ,
[UIElement.PreviewDragLeaveEvent](#) , [UIElement.PreviewDragOverEvent](#) ,
[UIElement.PreviewDropEvent](#) , [UIElement.PreviewGiveFeedbackEvent](#) ,
[UIElement.PreviewGotKeyboardFocusEvent](#) , [UIElement.PreviewKeyDownEvent](#) ,
[UIElement.PreviewKeyUpEvent](#) , [UIElement.PreviewLostKeyboardFocusEvent](#) ,
[UIElement.PreviewMouseDownEvent](#) , [UIElement.PreviewMouseLeftButtonDownEvent](#) ,
[UIElement.PreviewMouseLeftButtonUpEvent](#) , [UIElement.PreviewMouseMoveEvent](#) ,
[UIElement.PreviewMouseRightButtonDownEvent](#) , [UIElement.PreviewMouseRightButtonUpEvent](#) ,
[UIElement.PreviewMouseUpEvent](#) , [UIElement.PreviewMouseWheelEvent](#) ,
[UIElement.PreviewQueryContinueDragEvent](#) , [UIElement.PreviewStylusButtonDownEvent](#) ,
[UIElement.PreviewStylusButtonUpEvent](#) , [UIElement.PreviewStylusDownEvent](#) ,
[UIElement.PreviewStylusInAirMoveEvent](#) , [UIElement.PreviewStylusInRangeEvent](#) ,
[UIElement.PreviewStylusMoveEvent](#) , [UIElement.PreviewStylusOutOfRangeEvent](#) ,
[UIElement.PreviewStylusSystemGestureEvent](#) , [UIElement.PreviewStylusUpEvent](#) ,
[UIElement.PreviewTextInputEvent](#) , [UIElement.PreviewTouchDownEvent](#) ,
[UIElement.PreviewTouchMoveEvent](#) , [UIElement.PreviewTouchUpEvent](#) ,
[UIElement.QueryContinueDragEvent](#) , [UIElement.QueryCursorEvent](#) ,
[UIElement.RenderTransformOriginProperty](#) , [UIElement.RenderTransformProperty](#) ,
[UIElement.SnapsToDevicePixelsProperty](#) , [UIElement.StylusButtonDownEvent](#) ,
[UIElement.StylusButtonUpEvent](#) , [UIElement.StylusDownEvent](#) , [UIElement.StylusEnterEvent](#) ,
[UIElement.StylusInAirMoveEvent](#) , [UIElement.StylusInRangeEvent](#) , [UIElement.StylusLeaveEvent](#) ,
[UIElement.StylusMoveEvent](#) , [UIElement.StylusOutOfRangeEvent](#) ,
[UIElement.StylusSystemGestureEvent](#) , [UIElement.StylusUpEvent](#) , [UIElement.TextInputEvent](#) ,
[UIElement.TouchDownEvent](#) , [UIElement.TouchEnterEvent](#) , [UIElement.TouchLeaveEvent](#) ,
[UIElement.TouchMoveEvent](#) , [UIElement.TouchUpEvent](#) , [UIElement.UidProperty](#) ,
[UIElement.VisibilityProperty](#) , [UIElement.AddHandler\(RoutedEvent, Delegate\)](#) ,
[UIElement.AddHandler\(RoutedEvent, Delegate, bool\)](#) ,
[UIElement.AddToEventRoute\(EventRoute, RoutedEventArgs\)](#) ,
[UIElement.ApplyAnimationClock\(DependencyProperty, AnimationClock\)](#) ,
[UIElement.ApplyAnimationClock\(DependencyProperty, AnimationClock, HandoffBehavior\)](#) ,
[UIElement.Arrange\(Rect\)](#) , [UIElement.BeginAnimation\(DependencyProperty, AnimationTimeline\)](#) ,
[UIElement.BeginAnimation\(DependencyProperty, AnimationTimeline, HandoffBehavior\)](#) ,
[UIElement.CaptureMouse\(\)](#) , [UIElement.CaptureStylus\(\)](#) , [UIElement.CaptureTouch\(TouchDevice\)](#) ,
[UIElement.Focus\(\)](#) , [UIElement.GetAnimationBaseValue\(DependencyProperty\)](#) ,
[UIElement.HitTestCore\(GeometryHitTestParameters\)](#) , [UIElement.HitTestCore\(PointHitTestParameters\)](#) ,
[UIElement.InputHitTest\(Point\)](#) , [UIElement.InvalidateArrange\(\)](#) , [UIElement.InvalidateMeasure\(\)](#) ,
[UIElement.InvalidateVisual\(\)](#) , [UIElement.Measure\(Size\)](#) ,
[UIElement.OnAccessKey\(AccessKeyEventArgs\)](#) , [UIElement.OnChildDesiredSizeChanged\(UIElement\)](#) ,

[UIElement.OnDragEnter\(DragEventArgs\)](#) , [UIElement.OnDragLeave\(DragEventArgs\)](#) ,
[UIElement.OnDragOver\(DragEventArgs\)](#) , [UIElement.OnDrop\(DragEventArgs\)](#) ,
[UIElement.OnGiveFeedback\(GiveFeedbackEventArgs\)](#) ,
[UIElement.OnGotKeyboardFocus\(KeyboardFocusChangedEventArgs\)](#) ,
[UIElement.OnGotMouseCapture\(MouseEventArgs\)](#) ,
[UIElement.OnGotStylusCapture\(StylusEventArgs\)](#) , [UIElement.OnGotTouchCapture\(TouchEventArgs\)](#) ,
[UIElement.OnIsKeyboardFocusedChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsKeyboardFocusWithinChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsMouseCapturedChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsMouseCaptureWithinChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsMouseDirectlyOverChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsStylusCapturedChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsStylusCaptureWithinChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnIsStylusDirectlyOverChanged\(DependencyPropertyChangedEventArgs\)](#) ,
[UIElement.OnKeyDown\(KeyEventArgs\)](#) , [UIElement.OnKeyUp\(KeyEventArgs\)](#) ,
[UIElement.OnLostFocus\(RoutedEventArgs\)](#) ,
[UIElement.OnLostKeyboardFocus\(KeyboardFocusChangedEventArgs\)](#) ,
[UIElement.OnLostMouseCapture\(MouseEventArgs\)](#) ,
[UIElement.OnLostStylusCapture\(StylusEventArgs\)](#) , [UIElement.OnLostTouchCapture\(TouchEventArgs\)](#) ,
[UIElement.OnManipulationCompleted\(ManipulationCompletedEventArgs\)](#) ,
[UIElement.OnManipulationDelta\(ManipulationDeltaEventArgs\)](#) ,
[UIElement.OnManipulationInertiaStarting\(ManipulationInertiaStartingEventArgs\)](#) ,
[UIElement.OnManipulationStarted\(ManipulationStartedEventArgs\)](#) ,
[UIElement.OnManipulationStarting\(ManipulationStartingEventArgs\)](#) ,
[UIElement.OnMouseDown\(MouseButtonEventArgs\)](#) , [UIElement.OnMouseEnter\(MouseEventArgs\)](#) ,
[UIElement.OnMouseLeave\(MouseEventArgs\)](#) ,
[UIElement.OnMouseLeftButtonDown\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseLeftButtonUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseMove\(MouseEventArgs\)](#) ,
[UIElement.OnMouseRightButtonDown\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseRightButtonUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseUp\(MouseButtonEventArgs\)](#) ,
[UIElement.OnMouseWheel\(MouseEventArgs\)](#) ,
[UIElement.OnPreviewDragEnter\(DragEventArgs\)](#) , [UIElement.OnPreviewDragLeave\(DragEventArgs\)](#) ,
[UIElement.OnPreviewDragOver\(DragEventArgs\)](#) , [UIElement.OnPreviewDrop\(DragEventArgs\)](#) ,
[UIElement.OnPreviewGiveFeedback\(GiveFeedbackEventArgs\)](#) ,
[UIElement.OnPreviewGotKeyboardFocus\(KeyboardFocusChangedEventArgs\)](#) ,
[UIElement.OnPreviewKeyDown\(KeyEventArgs\)](#) , [UIElement.OnPreviewKeyUp\(KeyEventArgs\)](#) ,
[UIElement.OnPreviewLostKeyboardFocus\(KeyboardFocusChangedEventArgs\)](#) ,
[UIElement.OnPreviewMouseDown\(MouseButtonEventArgs\)](#) ,

[UIElement.OnPreviewMouseLeftButtonDown\(MouseEventArgs\)](#) ,
[UIElement.OnPreviewMouseLeftButtonUp\(MouseEventArgs\)](#) ,
[UIElement.OnPreviewMouseMove\(MouseEventArgs\)](#) ,
[UIElement.OnPreviewMouseRightButtonDown\(MouseEventArgs\)](#) ,
[UIElement.OnPreviewMouseRightButtonUp\(MouseEventArgs\)](#) ,
[UIElement.OnPreviewMouseUp\(MouseEventArgs\)](#) ,
[UIElement.OnPreviewMouseWheel\(MouseWheelEventArgs\)](#) ,
[UIElement.OnPreviewQueryContinueDrag\(QueryContinueDragEventArgs\)](#) ,
[UIElement.OnPreviewStylusButtonDown\(StylusButtonEventArgs\)](#) ,
[UIElement.OnPreviewStylusButtonUp\(StylusButtonEventArgs\)](#) ,
[UIElement.OnPreviewStylusDown\(StylusDownEventArgs\)](#) ,
[UIElement.OnPreviewStylusInAirMove\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewStylusInRange\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewStylusMove\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewStylusOutOfRange\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewStylusSystemGesture\(StylusSystemGestureEventArgs\)](#) ,
[UIElement.OnPreviewStylusUp\(StylusEventArgs\)](#) ,
[UIElement.OnPreviewTextInput\(TextCompositionEventArgs\)](#) ,
[UIElement.OnPreviewTouchDown\(TouchEventArgs\)](#) ,
[UIElement.OnPreviewTouchMove\(TouchEventArgs\)](#) , [UIElement.OnPreviewTouchUp\(TouchEventArgs\)](#) ,
[UIElement.OnQueryContinueDrag\(QueryContinueDragEventArgs\)](#) ,
[UIElement.OnQueryCursor\(QueryCursorEventArgs\)](#) , [UIElement.OnRender\(DrawingContext\)](#) ,
[UIElement.OnStylusButtonDown\(StylusButtonEventArgs\)](#) ,
[UIElement.OnStylusButtonUp\(StylusButtonEventArgs\)](#) ,
[UIElement.OnStylusDown\(StylusDownEventArgs\)](#) , [UIElement.OnStylusEnter\(StylusEventArgs\)](#) ,
[UIElement.OnStylusInAirMove\(StylusEventArgs\)](#) , [UIElement.OnStylusInRange\(StylusEventArgs\)](#) ,
[UIElement.OnStylusLeave\(StylusEventArgs\)](#) , [UIElement.OnStylusMove\(StylusEventArgs\)](#) ,
[UIElement.OnStylusOutOfRange\(StylusEventArgs\)](#) ,
[UIElement.OnStylusSystemGesture\(StylusSystemGestureEventArgs\)](#) ,
[UIElement.OnStylusUp\(StylusEventArgs\)](#) , [UIElement.OnTextInput\(TextCompositionEventArgs\)](#) ,
[UIElement.OnTouchDown\(TouchEventArgs\)](#) , [UIElement.OnTouchEnter\(TouchEventArgs\)](#) ,
[UIElement.OnTouchLeave\(TouchEventArgs\)](#) , [UIElement.OnTouchMove\(TouchEventArgs\)](#) ,
[UIElement.OnTouchUp\(TouchEventArgs\)](#) , [UIElement.RaiseEvent\(RoutedEventArgs\)](#) ,
[UIElement.ReleaseAllTouchCaptures\(\)](#) , [UIElement.ReleaseMouseCapture\(\)](#) ,
[UIElement.ReleaseStylusCapture\(\)](#) , [UIElement.ReleaseTouchCapture\(TouchDevice\)](#) ,
[UIElement.RemoveHandler\(RoutedEvent, Delegate\)](#) , [UIElement.TranslatePoint\(Point, UIElement\)](#) ,
[UIElement.UpdateLayout\(\)](#) , [UIElement.AllowDrop](#) , [UIElement.AreAnyTouchesCaptured](#) ,
[UIElement.AreAnyTouchesCapturedWithin](#) , [UIElement.AreAnyTouchesDirectlyOver](#) ,
[UIElement.AreAnyTouchesOver](#) , [UIElement.BitmapEffect](#) , [UIElement.BitmapEffectInput](#) ,
[UIElement.CacheMode](#) , [UIElement.Clip](#) , [UIElement.ClipToBounds](#) ,

[UIElement.CommandBindings](#) , [UIElement.DesiredSize](#) , [UIElement.Effect](#) , [UIElement.Focusable](#) ,
[UIElement.HasAnimatedProperties](#) , [UIElement.HasEffectiveKeyboardFocus](#) ,
[UIElement.InputBindings](#) , [UIElement.IsArrangeValid](#) , [UIElement.IsEnabled](#) ,
[UIElement.IsEnabledCore](#) , [UIElement.IsFocused](#) , [UIElement.IsHitTestVisible](#) ,
[UIElement.IsInputMethodEnabled](#) , [UIElement.IsKeyboardFocused](#) ,
[UIElement.IsKeyboardFocusWithin](#) , [UIElement.IsManipulationEnabled](#) , [UIElement.IsMeasureValid](#) ,
[UIElement.IsMouseCaptured](#) , [UIElement.IsMouseCaptureWithin](#) , [UIElement.IsMouseDirectlyOver](#) ,
[UIElement.IsMouseOver](#) , [UIElement.IsStylusCaptured](#) , [UIElement.IsStylusCaptureWithin](#) ,
[UIElement.IsStylusDirectlyOver](#) , [UIElement.IsStylusOver](#) , [UIElement.IsVisible](#) , [UIElement.Opacity](#) ,
[UIElement.OpacityMask](#) , [UIElement.PersistId](#) , [UIElement.RenderSize](#) ,
[UIElement.RenderTransform](#) , [UIElement.RenderTransformOrigin](#) , [UIElement.SnapsToDevicePixels](#) ,
[UIElement.StylusPlugIns](#) , [UIElement.TouchesCaptured](#) , [UIElement.TouchesCapturedWithin](#) ,
[UIElement.TouchesDirectlyOver](#) , [UIElement.TouchesOver](#) , [UIElement.Uid](#) , [UIElement.Visibility](#) ,
[UIElement.DragEnter](#) , [UIElement.DragLeave](#) , [UIElement.DragOver](#) , [UIElement.Drop](#) ,
[UIElement.FocusableChanged](#) , [UIElement.GiveFeedback](#) , [UIElement.GotFocus](#) ,
[UIElement.GotKeyboardFocus](#) , [UIElement.GotMouseCapture](#) , [UIElement.GotStylusCapture](#) ,
[UIElement.GetTouchCapture](#) , [UIElement.IsEnabledChanged](#) , [UIElement.IsHitTestVisibleChanged](#) ,
[UIElement.IsKeyboardFocusedChanged](#) , [UIElement.IsKeyboardFocusWithinChanged](#) ,
[UIElement.IsMouseCapturedChanged](#) , [UIElement.IsMouseCaptureWithinChanged](#) ,
[UIElement.IsMouseDirectlyOverChanged](#) , [UIElement.IsStylusCapturedChanged](#) ,
[UIElement.IsStylusCaptureWithinChanged](#) , [UIElement.IsStylusDirectlyOverChanged](#) ,
[UIElement.IsVisibleChanged](#) , [UIElement.KeyDown](#) , [UIElement.KeyUp](#) ,
[UIElement.LayoutUpdated](#) , [UIElement.LostFocus](#) , [UIElement.LostKeyboardFocus](#) ,
[UIElement.LostMouseCapture](#) , [UIElement.LostStylusCapture](#) , [UIElement.LostTouchCapture](#) ,
[UIElement.ManipulationBoundaryFeedback](#) , [UIElement.ManipulationCompleted](#) ,
[UIElement.ManipulationDelta](#) , [UIElement.ManipulationInertiaStarting](#) ,
[UIElement.ManipulationStarted](#) , [UIElement.ManipulationStarting](#) , [UIElement.MouseDown](#) ,
[UIElement.MouseEnter](#) , [UIElement.MouseLeave](#) , [UIElement.MouseLeftButtonDown](#) ,
[UIElement.MouseLeftButtonUp](#) , [UIElement.MouseMove](#) , [UIElement.MouseRightButtonDown](#) ,
[UIElement.MouseRightButtonUp](#) , [UIElement.MouseUp](#) , [UIElement.MouseWheel](#) ,
[UIElement.PreviewDragEnter](#) , [UIElement.PreviewDragLeave](#) , [UIElement.PreviewDragOver](#) ,
[UIElement.PreviewDrop](#) , [UIElement.PreviewGiveFeedback](#) , [UIElement.PreviewGotKeyboardFocus](#) ,
[UIElement.PreviewKeyDown](#) , [UIElement.PreviewKeyUp](#) , [UIElement.PreviewLostKeyboardFocus](#) ,
[UIElement.PreviewMouseDown](#) , [UIElement.PreviewMouseLeftButtonDown](#) ,
[UIElement.PreviewMouseLeftButtonUp](#) , [UIElement.PreviewMouseMove](#) ,
[UIElement.PreviewMouseRightButtonDown](#) , [UIElement.PreviewMouseRightButtonUp](#) ,
[UIElement.PreviewMouseUp](#) , [UIElement.PreviewMouseWheel](#) ,
[UIElement.PreviewQueryContinueDrag](#) , [UIElement.PreviewStylusButtonDown](#) ,
[UIElement.PreviewStylusButtonUp](#) , [UIElement.PreviewStylusDown](#) ,
[UIElement.PreviewStylusInAirMove](#) , [UIElement.PreviewStylusInRange](#) ,

[UIElement.PreviewStylusMove](#) , [UIElement.PreviewStylusOutOfRange](#) ,
[UIElement.PreviewStylusSystemGesture](#) , [UIElement.PreviewStylusUp](#) , [UIElement.PreviewTextInput](#) ,
[UIElement.PreviewTouchDown](#) , [UIElement.PreviewTouchMove](#) , [UIElement.PreviewTouchUp](#) ,
[UIElement.QueryContinueDrag](#) , [UIElement.QueryCursor](#) , [UIElement.StylusButtonDown](#) ,
[UIElement.StylusButtonUp](#) , [UIElement.StylusDown](#) , [UIElement.StylusEnter](#) ,
[UIElement.StylusInAirMove](#) , [UIElement.StylusInRange](#) , [UIElement.StylusLeave](#) ,
[UIElement.StylusMove](#) , [UIElement.StylusOutOfRange](#) , [UIElement.StylusSystemGesture](#) ,
[UIElement.StylusUp](#) , [UIElement.TextInput](#) , [UIElement.TouchDown](#) , [UIElement.TouchEnter](#) ,
[UIElement.TouchLeave](#) , [UIElement.TouchMove](#) , [UIElement.TouchUp](#) ,
[Visual.AddVisualChild\(Visual\)](#) , [Visual.FindCommonVisualAncestor\(DependencyObject\)](#) ,
[Visual.IsAncestorOf\(DependencyObject\)](#) , [Visual.IsDescendantOf\(DependencyObject\)](#) ,
[Visual.PointFromScreen\(Point\)](#) , [Visual.PointToScreen\(Point\)](#) , [Visual.RemoveVisualChild\(Visual\)](#) ,
[Visual.TransformToAncestor\(Visual3D\)](#) , [Visual.TransformToAncestor\(Visual\)](#) ,
[Visual.TransformToDescendant\(Visual\)](#) , [Visual.TransformToVisual\(Visual\)](#) , [Visual.VisualBitmapEffect](#) ,
[Visual.VisualBitmapEffectInput](#) , [Visual.VisualBitmapScalingMode](#) , [Visual.VisualCacheMode](#) ,
[Visual.VisualClearTypeHint](#) , [Visual.VisualClip](#) , [Visual.VisualEdgeMode](#) , [Visual.VisualEffect](#) ,
[Visual.VisualOffset](#) , [Visual.VisualOpacity](#) , [Visual.VisualOpacityMask](#) , [Visual.VisualParent](#) ,
[Visual.VisualScrollableAreaClip](#) , [Visual.VisualTextHintingMode](#) , [Visual.VisualTextRenderingMode](#) ,
[Visual.VisualTransform](#) , [Visual.VisualXSnappingGuidelines](#) , [Visual.VisualYSnappingGuidelines](#) ,
[DependencyObject.ClearValue\(DependencyProperty\)](#) ,
[DependencyObject.ClearValue\(DependencyPropertyKey\)](#) ,
[DependencyObject.CoerceValue\(DependencyProperty\)](#) , [DependencyObject.Equals\(object\)](#) ,
[DependencyObject.GetHashCode\(\)](#) , [DependencyObject.GetLocalValueEnumerator\(\)](#) ,
[DependencyObject.GetValue\(DependencyProperty\)](#) ,
[DependencyObject.InvalidateProperty\(DependencyProperty\)](#) ,
[DependencyObject.ReadLocalValue\(DependencyProperty\)](#) ,
[DependencyObject.SetCurrentValue\(DependencyProperty, object\)](#) ,
[DependencyObject.SetValue\(DependencyProperty, object\)](#) ,
[DependencyObject.SetValue\(DependencyPropertyKey, object\)](#) ,
[DependencyObject.ShouldSerializeProperty\(DependencyProperty\)](#) ,
[DependencyObject.DependencyObjectType](#) , [DependencyObject.IsSealed](#) ,
[DispatcherObject.Dispatcher](#) , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

`CanvasWindow()`

Konstruktor. Beachte '_canvasWindow' wurde im XAML-Text erklärt und definiert ! (x:Name="_canvasWindow"). Wird als Vergrößerungsfenster benutzt.

```
public CanvasWindow()
```

CanvasWindow(string)

Konstruktor. Beachte '_canvasWindow' wurde im XAML-Text erklärt und definiert ! (x:Name="_canvasWindow"). Wird als Vergrößerungsfenster benutzt.

```
public CanvasWindow(string titleText)
```

Parameters

titleText [string](#)

windows title

Fields

isNowToEnd

Flag, damit am echten Programmende auch wirklich das Fenster geschlossen wird.

```
public bool isNowToEnd
```

Field Value

[bool](#)

Methods

InitializeComponent()

InitializeComponent

```
public void InitializeComponent()
```

Struct ColorNumber

Namespace: [MatrixFFN.Tools](#)

Assembly: MatrixFFN.dll

helper for the colordnumbers used

```
public struct ColorNumber
```

Inherited Members

[ValueType.Equals\(object\)](#) , [ValueType.GetHashCode\(\)](#) , [ValueType.ToString\(\)](#) ,
[object.Equals\(object, object\)](#) , [object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Fields

Black

```
public static int Black
```

Field Value

[int](#)

Gray

```
public static int Gray
```

Field Value

[int](#)

Class DataNet

Namespace: [MatrixFFN.Tools](#)

Assembly: MatrixFFN.dll

Die Klasse 'DatenNetz' normalisiert die realen Daten und zurück.

Das geht für reale (Pattern) Features.

```
public class DataNet
```

Inheritance

[object](#) ← DataNet

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Remarks

Bei Java hatte ich einen MathContext benutzt - hier gibt es 'decimal'. Dieser Datentyp soll präzise aber langsam sein. Meine Theorie war bestimmt gut und sollte auch zukunftsweisend sein. Wichtig ist die Abbildung auf das Intervall und zurück.

Deswegen werde ich alles mit 'decimal' realisieren - dachte ich. Dieser Wertetyp wird aber nicht von 'Math' unterstützt und ist eine Baustelle. So muß es mit 'double' gehen.

Constructors

DataNet(BinaryReader)

Der Konstruktor, der sich aus der Speicherdatei lädt.

```
public DataNet(BinaryReader reader)
```

Parameters

reader [BinaryReader](#)

die übergebene offene Speicherdatei

DataNet(string)

Standardkonstruktor - wichtig für die Serialisierung.

```
public DataNet(string inName = "Eingabeseite")
```

Parameters

inName [string](#)

Fields

data

Datenobjekt der Klasse - alle Patterns werden hier gefunden.

```
public List<Pattern> data
```

Field Value

[List](#) <[Pattern](#)>

name

Positionsname: Eingabe- oder Ausgabeseite

```
public string name
```

Field Value

[string](#)

valuesAlpha

hat alle Lernwerte der Patterns

```
public List<double> valuesAlpha
```

Field Value

List <double>

valuesNormed

normierte Werte für das Netzwerk

```
public List<double> valuesNormed
```

Field Value

List <double>

valuesReal

reale weltliche Werte

```
public List<double> valuesReal
```

Field Value

List <double>

Methods

Add(Pattern)

Soll ein 'Pattern' der Liste hinzufügen.

```
public void Add(Pattern inPattern)
```

Parameters

inPattern [Pattern](#)

zu addierendes Pattern

Clear()

Setzt die Datenliste zurück.

```
public void Clear()
```

DataNetInit(double[][])

Hier werden automatisch aus den übergebenen Daten die Parameter fürs DatenNetz gebildet.

Bisher werden keine kategorischen Daten verarbeitet. (mögl. ToDo-Liste)

```
public Matrix DataNetInit(double[][] datenArray)
```

Parameters

datenArray [double](#)[][]

die Daten fürs Netz als Feld

Returns

[Matrix](#)

GetValuesAlpha()

Liefert den Alphavektor der die lokale Varianz des jeweiligen Patternobjektes ausliest. Dies ist für die angepasste Lernrate wichtig. Sollte für Ein- und Ausgabeschicht im Backpropverfahren benutzt werden.

```
public List<double> GetValuesAlpha()
```

Returns

[List](#) <[double](#)>

Ein Array der speziellen Lernwerte

LoadDataFromReader(BinaryReader)

eine traditionelle Laderoutine (binär)

```
public void LoadDataFromReader(BinaryReader reader)
```

Parameters

reader [BinaryReader](#)

ein BinaryReader

SaveDataToWriter(BinaryWriter)

eine traditionelle Speicherroutine (binär)

```
public void SaveDataToWriter(BinaryWriter writer)
```

Parameters

writer [BinaryWriter](#)

ein BinaryWriter

SetName(string)

Ich will das DatenNetz in der Ausgabe (ToString()) besser darstellen. Deswegen name = { Eingabeseite, Ausgabeseite }

Eingaben ungleich 'Eingabeseite' ergeben automatisch 'Ausgabeseite'

```
public void SetName(string inName = "Eingabeseite")
```

Parameters

inName string ↗

Size()

Liest die Größe der Datenliste aus und gibt sie zurück

`public int Size()`

Returns

int ↗

Größe der Datenliste

ToString()

Standardrepräsentation des Objekts.

`public override string ToString()`

Returns

string ↗

Beschreibung des DatenNetz'es

Class DataNetILGPU

Namespace: [MatrixFFN.Tools](#)

Assembly: MatrixFFN.dll

Die Klasse 'DatenNetz' normalisiert die realen Daten und zurück.

Das geht für reale (Pattern) Features.

```
public class DataNetILGPU
```

Inheritance

[object](#) ← DataNetILGPU

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Remarks

Bei Java hatte ich einen MathContext benutzt - hier gibt es 'decimal'. Dieser Datentyp soll präzise aber langsam sein. Meine Theorie war bestimmt gut und sollte auch zukunftsweisend sein. Wichtig ist die Abbildung auf das Intervall und zurück.

Deswegen werde ich alles mit 'decimal' realisieren - dachte ich. Dieser Wertetyp wird aber nicht von 'Math' unterstützt und ist eine Baustelle. So muß es mit 'double' gehen.

Constructors

DataNetILGPU(BinaryReader)

Der Konstruktor, der sich aus der Speicherdatei lädt.

```
public DataNetILGPU(BinaryReader reader)
```

Parameters

reader [BinaryReader](#)

die übergebene offene Speicherdatei

DataNetILGPU(string)

Standardkonstruktor - wichtig für die Serialisierung.

```
public DataNetILGPU(string inName = "Eingabeseite")
```

Parameters

inName [string](#)

Fields

data

Datenobjekt der Klasse - alle Patterns werden hier gefunden.

```
public List<Pattern> data
```

Field Value

[List](#) <[Pattern](#)>

name

Positionsname: Eingabe- oder Ausgabeseite

```
public string name
```

Field Value

[string](#)

valuesAlpha

hat alle Lernwerte der Patterns

```
public List<double> valuesAlpha
```

Field Value

[List](#)<[double](#)>

valuesNormed

normierte Werte für das Netzwerk

```
public List<double> valuesNormed
```

Field Value

[List](#)<[double](#)>

valuesReal

reale weltliche Werte

```
public List<double> valuesReal
```

Field Value

[List](#)<[double](#)>

Methods

Add(Pattern)

Soll ein 'Pattern' der Liste hinzufügen.

```
public void Add(Pattern inPattern)
```

Parameters

inPattern [Pattern](#)

zu addierendes Pattern

Clear()

Setzt die Datenliste zurück.

```
public void Clear()
```

DataNetInit(double[][][])

Hier werden automatisch aus den übergebenen Daten die Parameter fürs DatenNetz gebildet.

Bisher werden keine kategorischen Daten verarbeitet. (mögl. ToDo-Liste)

```
public MatrixILGPU DataNetInit(double[][][] datenArray)
```

Parameters

datenArray [double](#)[][]

die Daten fürs Netz als Feld

Returns

[MatrixILGPU](#)

GetValuesAlpha()

Liefert den Alphavektor der die lokale Varianz des jeweiligen Patternobjektes ausliest. Dies ist für die angepasste Lernrate wichtig. Sollte für Ein- und Ausgabeschicht im Backpropverfahren benutzt werden.

```
public List<double> GetValuesAlpha()
```

Returns

[List](#) <[double](#)>

Ein Array der speziellen Lernwerte

LoadDataFromReader(BinaryReader)

eine traditionelle Laderoutine (binär)

```
public void LoadDataFromReader(BinaryReader reader)
```

Parameters

reader [BinaryReader](#)

ein BinaryReader

SaveDataToWriter(BinaryWriter)

eine traditionelle Speicherroutine (binär)

```
public void SaveDataToWriter(BinaryWriter writer)
```

Parameters

writer [BinaryWriter](#)

ein BinaryWriter

SetName(string)

Ich will das DatenNetz in der Ausgabe (ToString()) besser darstellen. Deswegen name = { Eingabeseite, Ausgabeseite }

Eingaben ungleich 'Eingabeseite' ergeben automatisch 'Ausgabeseite'

```
public void SetName(string inName = "Eingabeseite")
```

Parameters

inName string ↗

Size()

Liest die Größe der Datenliste aus und gibt sie zurück

`public int Size()`

Returns

int ↗

Größe der Datenliste

ToString()

Standardrepräsentation des Objekts.

`public override string ToString()`

Returns

string ↗

Beschreibung des DatenNetz'es

Class Matrix

Namespace: [MatrixFFN.Tools](#)

Assembly: MatrixFFN.dll

Diese Klasse bildet Matrixrechenarten für das neuronale Netz ab.

Wichtig ist hierbei die Matrixmultiplikation von 2 Matrixen, die auch völlig korrekt nach Formelsammlung erfolgt!

Unterteilung der Funktionen in:

- Operation auf die interne Matrix
- static Matrix Funktion: Operation auf Rückgabematrix
- static bool Funktion: Operation auf Zielmatrix (beste Geschwindigkeit)

```
public class Matrix
```

Inheritance

[object](#) ← Matrix

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

Matrix()

Parameterloser Konstruktor für die Serialisierung.

```
public Matrix()
```

Matrix(BinaryReader)

Dieser Konstruktor lädt seine Daten aus dem BinaryReader und initialisiert sich damit.

```
public Matrix(BinaryReader reader)
```

Parameters

reader [BinaryReader](#)

ein BinaryReader

Matrix(int, int, double)

Konstruktor der Klasse, wobei ein bestimmter Wert für die Elemente gesetzt wird.

Am schnellsten ist die Nullmatrix erstellt.

```
public Matrix(int inX, int inY, double val = 0)
```

Parameters

inX [int](#)

die X-Größe der Matrix

inY [int](#)

die Y-Größe der Matrix

val [double](#)

bestimmter Elementwert

Matrix(int, int, double, double)

Konstruktor der Klasse, wobei die Matrix mit zufälligen Werten zwischen [-1, 1] gefüllt wird.

eigene Spannen sind erlaubt.

```
public Matrix(int inX, int inY, double min = -1, double max = 1)
```

Parameters

inx [int](#)

die X-Größe der Matrix

inY [int](#)

die Y-Größe der Matrix

min [double](#)

die untere Grenze für die Zufallszahlen

max [double](#)

die obere Grenze für die Zufallszahlen

Fields

data

Daten der Matrix

```
public double[,] data
```

Field Value

[double](#)[] ,]

fileName

Dateiname für das eigene Speichern

```
public string fileName
```

Field Value

[string](#)

sizeX

X-Größe der Matrix

```
public int sizeX
```

Field Value

[int](#)

sizeY

Y-Größe der Matrix

```
public int sizeY
```

Field Value

[int](#)

Methods

AddMatrix(Matrix)

Fügt eine gleichgroße Matrix zur internen hinzu.

```
public void AddMatrix(Matrix m)
```

Parameters

m [Matrix](#)

die hinzuzufügende Matrix

AddMatrix(Matrix, Matrix)

Hilfsfunktion zur Erleichterung. AddMatrixition zweier Matrixen.

```
public static Matrix AddMatrix(Matrix m1, Matrix m2)
```

Parameters

[m1 Matrix](#)

Matrix 1

[m2 Matrix](#)

Matrix 2

Returns

[Matrix](#)

Ergebnismatrix

AddMatrix(Matrix, Matrix, Matrix)

Hilfsfunktion zur Erleichterung. AddMatrixition zweier Matrixen.

```
public static void AddMatrix(Matrix m1, Matrix m2, Matrix target)
```

Parameters

[m1 Matrix](#)

Matrix 1

[m2 Matrix](#)

Matrix 2

[target Matrix](#)

Zielmatrix

AddScalar(Matrix, double)

Addiert den 'value' zu jedem einzelnen Matrixelement.

```
public static Matrix AddScalar(Matrix matrix, double value)
```

Parameters

matrix [Matrix](#)

Quellenmatrix

value [double](#) ↗

Wert zum Hinzufügen

Returns

[Matrix](#)

Ergebnismatrix

AddScalar(Matrix, double, Matrix)

Addiert den 'value' zu jedem einzelnen Matrixelement und liefert das Ergebnis an die Zielmatrix (beste Geschwindigkeit).

```
public static void AddScalar(Matrix source, double value, Matrix target)
```

Parameters

source [Matrix](#)

Quellenmatrix

value [double](#) ↗

Wert zum Hinzufügen

target [Matrix](#)

Erfolg der Operation

AddScalar(double)

Addiert den 'value' zu jedem einzelnen Element der internen Matrix.

```
public void AddScalar(double value)
```

Parameters

value [double](#)

Wert zum Hinzufügen

DeriveSigmoid()

Leitet die interne Matrix ab. Bringt nur nach 'ToSigmoid' Sinn.

```
public Matrix DeriveSigmoid()
```

Returns

[Matrix](#)

abgeleitete Matrix

DeriveSigmoid(Matrix)

Leitet die übergebene Matrix ab. Diese sollte per 'Sigmoid' erstellt sein.

```
public static Matrix DeriveSigmoid(Matrix sigMatrix)
```

Parameters

sigMatrix [Matrix](#)

Eingabematrix

Returns

Matrix

sigmoid abgeleitete Ergebnismatrix

DeriveSigmoid(Matrix, Matrix)

Leitet die übergebene Matrix ab. Diese sollte per 'Sigmoid' erstellt sein.

```
public static void DeriveSigmoid(Matrix sigMatrix, Matrix target)
```

Parameters

sigMatrix [Matrix](#)

Eingabematrix

target [Matrix](#)

sigmoid abgeleitete Ergebnismatrix

FromArray(double[])

Hilfsfunktion um zum Beispiel Eingabedaten in Netzwerkdaten umzuwandeln.

Diese Funktion erstellt eine [doubles.Length, 1]-Matrix

```
public static Matrix FromArray(double[] doubles)
```

Parameters

doubles [double](#)[]

das Feld mit den Werten

Returns

[Matrix](#)

FromArray(double[], Matrix)

Hilfsfunktion um zum Beispiel Eingabedaten in Netzwerkdaten umzuwandeln.

Diese Funktion erstellt eine [doubles.Length, 1]-Matrix

```
public static void FromArray(double[] doubles, Matrix target)
```

Parameters

doubles [double\[\]](#)

das Feld mit den Werten

target [Matrix](#)

die Zielmatrix

FromArrayTranspose(double[])

Hilfsfunktion um zum Beispiel Eingabedaten in Netzwerkdaten umzuwandeln.

Diese Funktion erstellt eine [1, doubles.Length]-Transpose-Matrix

```
public static Matrix FromArrayTranspose(double[] doubles)
```

Parameters

doubles [double\[\]](#)

das Feld mit den Werten

Returns

[Matrix](#)

FromArrayTranspose(double[], Matrix)

Hilfsfunktion um zum Beispiel Eingabedaten in Netzwerkdaten umzuwandeln.

Diese Funktion erstellt eine [1, doubles.Length]-Transpose-Matrix

```
public static void FromArrayTranspose(double[] doubles, Matrix target)
```

Parameters

doubles [double](#)[]

das Feld mit den Werten

target [Matrix](#)

Zielmatrix

LoadDataFromReader(BinaryReader)

eine traditionelle Laderoutine (binär)

```
public void LoadDataFromReader(BinaryReader reader)
```

Parameters

reader [BinaryReader](#)

ein BinaryReader

LoadMatrixLocal()

Lädt die Matrix aus eigener Datei. ('public string fileName' !)

```
public void LoadMatrixLocal()
```

MS_Sum()

Wurzel aus der Elementsumme (mean square sum) für die Fehlerberechnung.

```
public double MS_Sum()
```

Returns

double ↗

der absolute Wert der Summe

Multiply(Matrix, Matrix)

Das Herzstück: ermittelt das 'Kreuzprodukt' beider Matrixen.

Beachte: (m1.sizeX == m2.sizeY) -> target[m2.sizeX, m1.sizeY]

```
public static Matrix Multiply(Matrix m1, Matrix m2)
```

Parameters

m1 [Matrix](#)

Eingabematrix 1

m2 [Matrix](#)

Eingabematrix 2

Returns

[Matrix](#)

Ergebnismatrix

Multiply(Matrix, Matrix, Matrix)

Das Herzstück: ermittelt das 'Kreuzprodukt' beider Matrixen.

Beachte: (m1.sizeX == m2.sizeY) -> target[m2.sizeX, m1.sizeY]

```
public static void Multiply(Matrix m1, Matrix m2, Matrix target)
```

Parameters

[m1 Matrix](#)

Eingabematrix 1

[m2 Matrix](#)

Eingabematrix 2

[target Matrix](#)

Zielmatrix

MultiplySameSize(Matrix)

Multipliziert die Eingabematrix zur internen falls sie die gleiche Größe hat.

```
public void MultiplySameSize(Matrix m)
```

Parameters

[m Matrix](#)

die Eingabematix

MultiplySameSize(Matrix, Matrix)

Multipliziert die Eingabematrixen falls sie die gleiche Größe haben.

```
public static Matrix MultiplySameSize(Matrix m1, Matrix m2)
```

Parameters

[m1 Matrix](#)

die 1.Eingabematix

[m2 Matrix](#)

die 2.Ringabematrix

Returns

[Matrix](#)

die Ergebnismatrix

MultiplySameSize(Matrix, Matrix, Matrix)

Multipliziert die Eingabematrixen falls sie die gleiche Größe haben. Ergebnis wird in die Zielmatrix geschrieben (beste Geschwindigkeit).

```
public static void MultiplySameSize(Matrix m1, Matrix m2, Matrix target)
```

Parameters

[m1 Matrix](#)

die 1.EIngabematix

[m2 Matrix](#)

die 2.Ringabematrix

[target Matrix](#)

Zielmatrix

MultiplyScalar(Matrix, double)

Multipliziert jedes Matrixelement mit einem Wert.

```
public static Matrix MultiplyScalar(Matrix source, double value)
```

Parameters

[source Matrix](#)

Quellmatrix

[value double](#)

der Multiplikationswert

Returns

[Matrix](#)

Ergebnismatrix

MultiplyScalar(Matrix, double, Matrix)

Multipliziert jedes Matrixelement mit einem Wert und liefert das Ergebnis in die Zielmatrix (beste Geschwindigkeit).

```
public static void MultiplyScalar(Matrix source, double value, Matrix target)
```

Parameters

source [Matrix](#)

Quellmatrix

value [double](#) ↗

der Multiplikationswert

target [Matrix](#)

Zielmatrix

MultiplyScalar(double)

Multipliziert jedes Matrixelement mit einem Wert.

```
public void MultiplyScalar(double value)
```

Parameters

value [double](#) ↗

der Multiplikationswert

Print()

Gibt die Matrix per ToString() an die Standardausgabe aus.

```
public void Print()
```

SaveDataToWriter(BinaryWriter)

eine traditionelle Speicherroutine (binär)

```
public void SaveDataToWriter(BinaryWriter writer)
```

Parameters

writer [BinaryWriter](#)

ein BinaryWriter

SaveMatrixLocal()

Speichert die Matrix mit eigener Datei. ('public string fileName' !)

```
public void SaveMatrixLocal()
```

SubtractMatrix(Matrix)

Zieht eine gleichgroße Matrix von der internen ab.

```
public void SubtractMatrix(Matrix m)
```

Parameters

m [Matrix](#)

die abzuziehende Matrix

SubtractMatrix(Matrix, Matrix)

Hilfsfunktion zur Erleichterung. Subtraktion zweier Matrixen.

```
public static Matrix SubtractMatrix(Matrix m1, Matrix m2)
```

Parameters

[m1 Matrix](#)

Matrix 1

[m2 Matrix](#)

Matrix 2

Returns

[Matrix](#)

Ergebnismatrix

SubtractMatrix(Matrix, Matrix, Matrix)

Hilfsfunktion zur Erleichterung. Subtrahiert zwei Matrixen und schreibt das Ergebnis in die Zielmatrix (beste Geschwindigkeit).

```
public static void SubtractMatrix(Matrix m1, Matrix m2, Matrix target)
```

Parameters

[m1 Matrix](#)

Matrix 1

[m2 Matrix](#)

Matrix 2

[target Matrix](#)

Zielmatrix

SubtractScalar(Matrix, double)

Subtrahiert den 'value' von jedem einzelnen Matrixelement.

```
public static Matrix SubtractScalar(Matrix matrix, double value)
```

Parameters

matrix [Matrix](#)

Quellenmatrix

value [double](#) ↗

Wert zum Abziehen

Returns

[Matrix](#)

Ergebnismatrix

SubtractScalar(Matrix, double, Matrix)

Subtrahiert den 'value' von jedem einzelnen Matrixelement und liefert das Ergebnis an die Zielmatrix (beste Geschwindigkeit).

```
public static void SubtractScalar(Matrix source, double value, Matrix target)
```

Parameters

source [Matrix](#)

Quellenmatrix

value [double](#) ↗

Wert zum Abziehen

target [Matrix](#)

Ziel der Operation

SubtractScalar(double)

Subtrahiert den 'value' von jedem einzelnen Element der internen Matrix.

```
public void SubtractScalar(double value)
```

Parameters

value double[]

Wert zum Abziehen

ToArrayList()

Schreibt den Inhalt der internen Matrix in ein Feld.

```
public double[] ToArrayList()
```

Returns

double[][]

das Ergebnisfeld

ToSigmoid()

Wendet die unsymmetrische logistische Funktion auf die interne Matrix an - verändert sie also.

```
public void ToSigmoid()
```

ToSigmoid(Matrix)

Liefert das Ergebnis der unsymmetrischen logistischen Funktion zurück.

```
public static Matrix ToSigmoid(Matrix source)
```

Parameters

source [Matrix](#)

die Eingabematrix

Returns

[Matrix](#)

die sigmoide Ergebnismatrix

ToSigmoid(Matrix, Matrix)

Liefert das Ergebnis der unsymmetrischen logistischen Funktion im 'target' (beste Geschwindigkeit).

```
public static void ToSigmoid(Matrix source, Matrix target)
```

Parameters

source [Matrix](#)

die Eingabematrix

target [Matrix](#)

die Zielmatrix

ToString()

Standardausgabe der Matrix.

```
public override string ToString()
```

Returns

[string](#) ↗

Stringrepräsentation der Matrix

Transpose(Matrix)

Transponiert die Matrix.

```
public static Matrix Transpose(Matrix m)
```

Parameters

m [Matrix](#)

Eingabematrix

Returns

[Matrix](#)

Ergebnismatrix [m.sizeY, m.sizeX]

Transpose(Matrix, Matrix)

Transponiert die Eingabematrix in die Zielmatrix (beste Geschwindigkeit).

```
public static void Transpose(Matrix source, Matrix target)
```

Parameters

source [Matrix](#)

Eingabematrix

target [Matrix](#)

Zielmatrix

Class MatrixILGPU

Namespace: [MatrixFFN.Tools](#)

Assembly: MatrixFFN.dll

Diese Klasse bildet Matrixrechenarten für das neuronale Netz ab. Variante für ILGPU.

Wichtig ist hierbei die Matrixmultiplikation von 2 Matrixen, die auch völlig korrekt nach Formelsammlung erfolgt!

ILGPU braucht zusätzlich zur Logik die GPU-Seite, bei der über Aktionen die jeweiligen Kernel ausgeführt werden.

Unterteilung der Funktionen in:

- Operation auf die interne Matrix
- static Matrix Funktion: Operation auf Rückgabematrix
- static void Funktion: Operation auf Zielmatrix (beste Geschwindigkeit)

```
public class MatrixILGPU
```

Inheritance

[object](#) ← MatrixILGPU

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

MatrixILGPU()

Parameterloser Konstruktor für die Serialisierung.

```
public MatrixILGPU()
```

MatrixILGPU(BinaryReader)

Dieser Konstruktor lädt seine Daten aus dem BinaryReader und initialisiert sich damit.

```
public MatrixILGPU(BinaryReader reader)
```

Parameters

reader [BinaryReader](#)

ein BinaryReader

MatrixILGPU(int, int, double)

Konstruktor der Klasse, wobei ein bestimmter Wert für die Elemente gesetzt wird.

Am schnellsten ist die Nullmatrix erstellt.

```
public MatrixILGPU(int inX, int inY, double val = 0)
```

Parameters

inX [int](#)

die X-Größe der Matrix

inY [int](#)

die Y-Größe der Matrix

val [double](#)

bestimmter Elementwert

MatrixILGPU(int, int, double, double)

Konstruktor der Klasse, wobei die Matrix mit zufälligen Werten zwischen [min, max] gefüllt wird.

eigene Spannen sind erlaubt.

```
public MatrixILGPU(int inX, int inY, double min, double max)
```

Parameters

inX [int](#)

die X-Größe der Matrix

inY [int](#)

die Y-Größe der Matrix

min [double](#)

die untere Grenze für die Zufallszahlen

max [double](#)

die obere Grenze für die Zufallszahlen

Fields

data

Daten der Matrix

```
public double[,] data
```

Field Value

[double](#)[]

fileName

Dateiname für das eigene Speichern

```
public string fileName
```

Field Value

[string](#) ↗

sizeX

X-Größe der Matrix

```
public int sizeX
```

Field Value

[int](#) ↗

sizeY

Y-Größe der Matrix

```
public int sizeY
```

Field Value

[int](#) ↗

Methods

AddMatrix(MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus.

Inhaltlich: Fügt eine gleichgroße Matrix zur internen hinzu.

```
public void AddMatrix(MatrixILGPU m)
```

Parameters

m [MatrixILGPU](#)

die hinzuzufügende Matrix

AddMatrix(MatrixILGPU, MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus.

Inhaltlich: Hilfsfunktion zur Erleichterung. Addition zweier Matrixen.

```
public static MatrixILGPU AddMatrix(MatrixILGPU m1, MatrixILGPU m2)
```

Parameters

m1 [MatrixILGPU](#)

Matrix 1

m2 [MatrixILGPU](#)

Matrix 2

Returns

[MatrixILGPU](#)

Ergebnismatrix

AddMatrix_instance_Kernel(Index2D, ArrayView2D<double, DenseX>, ArrayView2D<double, DenseX>)

ILGPU: Kernel für die Funktion

```
public static void AddMatrix_instance_Kernel(Index2D index, ArrayView2D<double, Stride2D.DenseX> intern, ArrayView2D<double, Stride2D.DenseX> source)
```

Parameters

index Index2D

"2D-Position in der Matrix

intern ArrayView2D<[double](#), Stride2D.DenseX>

die interne Matrix

source ArrayView2D<[double](#), Stride2D.DenseX>

Quellmatrix

AddMatrix_static_Kernel(Index2D, ArrayView2D<double, DenseX>, ArrayView2D<double, DenseX>, ArrayView2D<double, DenseX>)

ILGPU: Kernel für die Funktion

```
public static void AddMatrix_static_Kernel(Index2D index, ArrayView2D<double, Stride2D.DenseX> source1, ArrayView2D<double, Stride2D.DenseX> source2, ArrayView2D<double, Stride2D.DenseX> target)
```

Parameters

index Index2D

"2D-Position in der Matrix

source1 ArrayView2D<[double](#), Stride2D.DenseX>

Quellmatrix 1

source2 ArrayView2D<[double](#), Stride2D.DenseX>

Quellmatrix2

target ArrayView2D<[double](#), Stride2D.DenseX>

Zielmatrix

AddMatrix_target(MatrixILGPU, MatrixILGPU, MatrixILGPU)

Hilfsfunktion zur Erleichterung. Addition zweier Matrixen.

```
public static void AddMatrix_target(MatrixILGPU m1, MatrixILGPU m2, MatrixILGPU target)
```

Parameters

m1 [MatrixILGPU](#)

Matrix 1

m2 [MatrixILGPU](#)

Matrix 2

target [MatrixILGPU](#)

Zielmatrix

AddScalar(MatrixILGPU, double)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: Addiert den 'value' zu jedem einzelnen Matrixelement.

```
public static MatrixILGPU AddScalar(MatrixILGPU matrix, double value)
```

Parameters

matrix [MatrixILGPU](#)

Quellenmatrix

value [double](#)

Wert zum Hinzufügen

Returns

[MatrixILGPU](#)

Ergebnismatrix

AddScalar(double)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus.

Inhaltlich: Addiert den 'value' zu jedem einzelnen Element der internen Matrix.

```
public void AddScalar(double value)
```

Parameters

value [double](#)

Wert zum Hinzufügen

AddScalar_Target(MatrixILGPU, double, MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: Addiert den 'value' zu jedem einzelnen Matrixelement und liefert das Ergebnis an die Zielmatrix (beste Geschwindigkeit).

```
public static void AddScalar_Target(MatrixILGPU source, double value, MatrixILGPU target)
```

Parameters

source [MatrixILGPU](#)

Quellenmatrix

value [double](#)

Wert zum Hinzufügen

target [MatrixILGPU](#)

Zielmatrix

DeriveSigmoid()

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: leitet die interne Matrix ab. Bringt nur nach 'ToSigmoid' Sinn.

```
public MatrixILGPU DeriveSigmoid()
```

Returns

[MatrixILGPU](#)

abgeleitete Matrix

DeriveSigmoid(MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: leitet die übergebene Matrix ab. Diese sollte per 'Sigmoid' erstellt sein.

```
public static MatrixILGPU DeriveSigmoid(MatrixILGPU sigMatrix)
```

Parameters

sigMatrix [MatrixILGPU](#)

Eingabematrix

Returns

[MatrixILGPU](#)

sigmoid abgeleitete Ergebnismatrix

DeriveSigmoid(MatrixILGPU, MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: leitet die übergebene Matrix ab. Diese sollte per 'Sigmoid' erstellt sein. Ergebnis wird in die Zielmatrix geschrieben (beste Geschwindigkeit).

```
public static void DeriveSigmoid(MatrixILGPU sigMatrix, MatrixILGPU target)
```

Parameters

sigMatrix [MatrixILGPU](#)

Eingabematrix

target [MatrixILGPU](#)

sigmoid abgeleitete Ergebnismatrix

DeriveSigmoid_any_Kernel(Index2D, ArrayView2D<double, DenseX>, ArrayView2D<double, DenseX>)

ILGPU: Kernel für die Funktion

```
public static void DeriveSigmoid_any_Kernel(Index2D index, ArrayView2D<double, Stride2D.DenseX> source, ArrayView2D<double, Stride2D.DenseX> target)
```

Parameters

index Index2D

Position im 2D-Feld

source ArrayView2D<[double](#)>, Stride2D.DenseX>

Quellmatrix

target ArrayView2D<[double](#)>, Stride2D.DenseX>

Zielmatrix

FromArray(double[])

Hilfsfunktion um zum Beispiel Eingabedaten in Netzwerkdaten umzuwandeln.

Diese Funktion erstellt eine [doubles.Length, 1]-Matrix

```
public static MatrixILGPU FromArray(double[] doubles)
```

Parameters

`doubles double[]`

das Feld mit den Werten

Returns

[MatrixILGPU](#)

FromArray(double[], MatrixILGPU)

Hilfsfunktion um zum Beispiel Eingabedaten in Netzwerkdaten umzuwandeln.

Diese Funktion erstellt eine [doubles.Length, 1]-Matrix

```
public static void FromArray(double[] doubles, MatrixILGPU target)
```

Parameters

`doubles double[]`

das Feld mit den Werten

`target MatrixILGPU`

die Zielmatrix

FromArrayTranspose(double[])

Hilfsfunktion um zum Beispiel Eingabedaten in Netzwerkdaten umzuwandeln.

Diese Funktion erstellt eine [1, doubles.Length]-Transpose-Matrix

```
public static MatrixILGPU FromArrayTranspose(double[] doubles)
```

Parameters

`doubles double[]`

das Feld mit den Werten

Returns

[MatrixILGPU](#)

Transponierter Array als Matrix

FromArrayTranspose(double[], MatrixILGPU)

Hilfsfunktion um zum Beispiel Eingabedaten in Netzwerkdaten umzuwandeln.

Diese Funktion erstellt eine [1, doubles.Length]-Transpose-Matrix

```
public static void FromArrayTranspose(double[] doubles, MatrixILGPU target)
```

Parameters

doubles [double](#)[]

das Feld mit den Werten

target [MatrixILGPU](#)

Zielmatrix

LoadDataFromReader(BinaryReader)

eine traditionelle Laderoutine (binär): benutzt den übergebenen 'BinaryReader'

```
public void LoadDataFromReader(BinaryReader reader)
```

Parameters

reader [BinaryReader](#)[]

ein BinaryReader

LoadMatrixLocal()

Lädt die Matrix aus eigener Datei. ('public string fileName' !)

```
public void LoadMatrixLocal()
```

MS_Sum()

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: Wurzel aus der Elementsumme (mean square sum) für die Fehlerberechnung.

```
public double MS_Sum()
```

Returns

double

der absolute Wert der Summe

MS_Sum(MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: Wurzel aus der Elementsumme (mean square sum) für die Fehlerberechnung.

```
public static double MS_Sum(MatrixILGPU matrix)
```

Parameters

matrix MatrixILGPU

Returns

double

der absolute Wert der Summe

MS_Sum(MatrixILGPU, double)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: Wurzel aus der Elementsumme (mean square sum) für die Fehlerberechnung.

```
public static void MS_Sum(MatrixILGPU matrix, double summe)
```

Parameters

matrix [MatrixILGPU](#)

summe [double](#)

MeanSquare_any_Kernel(Index2D, ArrayView2D<double, DenseX>, double)

ILGPU: Kernel für die Funktion

```
public static void MeanSquare_any_Kernel(Index2D index, ArrayView2D<double, Stride2D.DenseX> matrix, double summe)
```

Parameters

index Index2D

Position im 2D-Feld

matrix ArrayView2D<[double](#), Stride2D.DenseX>

Quellmatrix

summe [double](#)

mean square summe

Multiply(MatrixILGPU, MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: das Herzstück -> ermittelt das 'Kreuzprodukt' beider Matrixen.

Beachte: (m1.sizeX == m2.sizeY) -> target[m2.sizeX, m1.sizeY]

```
public static MatrixILGPU Multiply(MatrixILGPU m1, MatrixILGPU m2)
```

Parameters

m1 [MatrixILGPU](#)

Eingabematrix 1

m2 [MatrixILGPU](#)

Eingabematrix 2

Returns

[MatrixILGPU](#)

Ergebnismatrix

Multiply(MatrixILGPU, MatrixILGPU, MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: das Herzstück -> ermittelt das 'Kreuzprodukt' beider Matrixen. Ergebnis wird in die Zielmatrix geschrieben (beste Geschwindigkeit).

Beachte: (m1.sizeX == m2.sizeY) -> target[m2.sizeX, m1.sizeY]

```
public static void Multiply(MatrixILGPU m1, MatrixILGPU m2, MatrixILGPU target)
```

Parameters

m1 [MatrixILGPU](#)

Eingabematrix 1

[m2 MatrixILGPU](#)

Eingabematrix 2

[target MatrixILGPU](#)

Zielmatrix

MultiplySameSize(MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: Multipliziert die Eingabematrix zur internen falls sie die gleiche Größe hat.

```
public void MultiplySameSize(MatrixILGPU m)
```

Parameters

[m MatrixILGPU](#)

die Eingabematix

MultiplySameSize(MatrixILGPU, MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: Multipliziert die Eingabematrixen falls sie die gleiche Größe haben.

```
public static MatrixILGPU MultiplySameSize(MatrixILGPU m1, MatrixILGPU m2)
```

Parameters

[m1 MatrixILGPU](#)

die 1.Eingabematix

[m2 MatrixILGPU](#)

die 2.Ringabematrix

Returns

[MatrixILGPU](#)

die Ergebnismatrix

MultiplySameSize(MatrixILGPU, MatrixILGPU, MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: Multipliziert die Eingabematrixen falls sie die gleiche Größe haben. Ergebnis wird in die Zielmatrix geschrieben (beste Geschwindigkeit).

```
public static void MultiplySameSize(MatrixILGPU m1, MatrixILGPU m2, MatrixILGPU target)
```

Parameters

[m1 MatrixILGPU](#)

die 1.Eingabematix

[m2 MatrixILGPU](#)

die 2.Ringabematix

[target MatrixILGPU](#)

Zielmatrix

MultiplySameSize_instance_Kernel(Index2D, ArrayView2D<double, DenseX>, ArrayView2D<double, DenseX>)

ILGPU: Kernel für die Funktion

```
public static void MultiplySameSize_instance_Kernel(Index2D index, ArrayView2D<double, Stride2D.DenseX> intern, ArrayView2D<double, Stride2D.DenseX> source)
```

Parameters

index Index2D

Position in der Zielmatrix

intern ArrayView2D<[double](#), Stride2D.DenseX>

interne Zielmatrix

source ArrayView2D<[double](#), Stride2D.DenseX>

Quellmatrix

MultiplySameSize_static_Kernel(Index2D, ArrayView2D<double, DenseX>, ArrayView2D<double, DenseX>, ArrayView2D<double, DenseX>)

ILGPU: Kernel für die Funktion

```
public static void MultiplySameSize_static_Kernel(Index2D index, ArrayView2D<double, Stride2D.DenseX> source1, ArrayView2D<double, Stride2D.DenseX> source2, ArrayView2D<double, Stride2D.DenseX> target)
```

Parameters

index Index2D

Position im 2D-Feld

source1 ArrayView2D<[double](#), Stride2D.DenseX>

Quellmatrix 1

source2 ArrayView2D<[double](#), Stride2D.DenseX>

Quellmatrix 2

target ArrayView2D<[double](#), Stride2D.DenseX>

Zielmatrix

MultiplyScalar(MatrixILGPU, double)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: Multipliziert jedes Matrixelement mit einem Wert. Statische Funktion, die eine neue Datenklasse als Ergebnis liefert.

```
public static MatrixILGPU MultiplyScalar(MatrixILGPU source, double value)
```

Parameters

source [MatrixILGPU](#)

Quellmatrix

value [double](#) ↗

der Multiplikationswert

Returns

[MatrixILGPU](#)

Ergebnismatrix

MultiplyScalar(MatrixILGPU, double, MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: Multipliziert jedes Matrixelement mit einem Wert und liefert das Ergebnis in die Zielmatrix (beste Geschwindigkeit).

```
public static void MultiplyScalar(MatrixILGPU source, double value, MatrixILGPU target)
```

Parameters

source [MatrixILGPU](#)

Quellmatrix

value [double](#) ↗

der Multiplikationswert

target [MatrixILGPU](#)

Zielmatrix

MultiplyScalar(double)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: Multipliziert jedes Matrixelement mit einem Wert.

```
public void MultiplyScalar(double value)
```

Parameters

value [double](#)

der Multiplikationswert

MultiplyScalar_instance_Kernel(Index2D, ArrayView2D<double, DenseX>, double)

ILGPU: Kernel für die Funktion

```
public static void MultiplyScalar_instance_Kernel(Index2D index, ArrayView2D<double, Stride2D.DenseX> intern, double value)
```

Parameters

index Index2D

Position in der Zielmatrix

intern ArrayView2D<[double](#), Stride2D.DenseX>

Zielmatrix ist die interne Instanzmatrix

value [double](#)

Skalarwert für die Multiplikation

MultiplyScalar_static_Kernel(Index2D, ArrayView2D<double, DenseX>, double, ArrayView2D<double, DenseX>)

ILGPU: Kernel für die Funktion

```
public static void MultiplyScalar_static_Kernel(Index2D index, ArrayView2D<double, Stride2D.DenseX> source, double value, ArrayView2D<double>, Stride2D.DenseX> target)
```

Parameters

index Index2D

Position in der Zielmatrix

source ArrayView2D<[double](#)>, Stride2D.DenseX>

Quellmatrix

value [double](#)

Skalarwert für die Multiplikation

target ArrayView2D<[double](#)>, Stride2D.DenseX>

Zielmatrix

Multiply_static_Kernel(Index2D, ArrayView2D<double, DenseX>, ArrayView2D<double, DenseX>, int, ArrayView2D<double, DenseX>)

ILGPU: Kernel für die Funktion

```
public static void Multiply_static_Kernel(Index2D index, ArrayView2D<double, Stride2D.DenseX> m1, ArrayView2D<double>, Stride2D.DenseX> m2, int kante, ArrayView2D<double, Stride2D.DenseX> target)
```

Parameters

index Index2D

Position im 2D-Feld

m1 ArrayView2D<[double](#), Stride2D.DenseX>

erste Matrix

m2 ArrayView2D<[double](#), Stride2D.DenseX>

zweite Matrix

kante [int](#)

Länge der gleichen Seite

target ArrayView2D<[double](#), Stride2D.DenseX>

Zielmatrix

Print()

Gibt die Matrix per ToString() an die Standardausgabe aus.

```
public void Print()
```

SaveDataToWriter(BinaryWriter)

eine traditionelle Speicherroutine (binär): benutzt den übergebenen 'BinaryWriter'

```
public void SaveDataToWriter(BinaryWriter writer)
```

Parameters

writer [BinaryWriter](#)

ein BinaryWriter

SaveMatrixLocal()

Speichert die Matrix mit eigener Datei. ('public string fileName' !)

```
public void SaveMatrixLocal()
```

SubtractMatrix(MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: zieht eine gleichgroße Matrix von der internen ab.

```
public void SubtractMatrix(MatrixILGPU m)
```

Parameters

m [MatrixILGPU](#)

die abzuziehende Matrix

SubtractMatrix(MatrixILGPU, MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: Hilfsfunktion zur Erleichterung. Subtraktion zweier Matrixen.

```
public static MatrixILGPU SubtractMatrix(MatrixILGPU m1, MatrixILGPU m2)
```

Parameters

m1 [MatrixILGPU](#)

Matrix 1

m2 [MatrixILGPU](#)

Matrix 2

Returns

[MatrixILGPU](#)

SubtractMatrix(MatrixILGPU, MatrixILGPU, MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: Hilfsfunktion zur Erleichterung. Subtrahiert zwei Matrixen und schreibt das Ergebnis in die Zielmatrix (beste Geschwindigkeit).

```
public static void SubtractMatrix(MatrixILGPU m1, MatrixILGPU m2, MatrixILGPU target)
```

Parameters

m1 [MatrixILGPU](#)

Matrix 1

m2 [MatrixILGPU](#)

Matrix 2

target [MatrixILGPU](#)

Zielmatrix

SubtractMatrix_instance_Kernel(Index2D, ArrayView2D<double, DenseX>, ArrayView2D<double, DenseX>)

ILGPU: Kernel für die Funktion

```
public static void SubtractMatrix_instance_Kernel(Index2D index, ArrayView2D<double, Stride2D.DenseX> intern, ArrayView2D<double, Stride2D.DenseX> source)
```

Parameters

index Index2D

"2D-Position in der Matrix

intern [ArrayView2D<double>](#), Stride2D.DenseX>

Zielmatrix

source ArrayView2D<[double](#), Stride2D.DenseX>

Quellmatrix

SubtractMatrix_static_Kernel(Index2D, ArrayView2D<double, DenseX>, ArrayView2D<double, DenseX>, ArrayView2D<double, DenseX>)

ILGPU: Kernel für die Funktion

```
public static void SubtractMatrix_static_Kernel(Index2D index, ArrayView2D<double, Stride2D.DenseX> source1, ArrayView2D<double, Stride2D.DenseX> source2, ArrayView2D<double, Stride2D.DenseX> target)
```

Parameters

index Index2D

"2D-Position in der Matrix

source1 ArrayView2D<[double](#), Stride2D.DenseX>

Zielmatrix

source2 ArrayView2D<[double](#), Stride2D.DenseX>

Quellmatrix

target ArrayView2D<[double](#), Stride2D.DenseX>

Zielmatrix

SubtractScalar(MatrixILGPU, double)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben. (Hier wird der Additionskernel benutzt.)

Inhaltlich: Subtrahiert den 'value' von jedem einzelnen Matrixelement.

```
public static MatrixILGPU SubtractScalar(MatrixILGPU matrix, double value)
```

Parameters

matrix [MatrixILGPU](#)

Quellenmatrix

value [double](#)

Wert zum Abziehen

Returns

[MatrixILGPU](#)

Ergebnismatrix

SubtractScalar(double)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. (Hier wird der Additionskernel benutzt.)

Inhaltlich: Subtrahiert den 'value' von jedem einzelnen Element der internen Matrix.

```
public void SubtractScalar(double value)
```

Parameters

value [double](#)

Wert zum Abziehen

SubtractScalar_Target(MatrixILGPU, double, MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben. (Hier wird der Additionskernel benutzt.)

Subtrahiert den 'value' von jedem einzelnen Matrixelement und liefert das Ergebnis an die Zielmatrix (beste Geschwindigkeit).

```
public static void SubtractScalar_Target(MatrixILGPU source, double value,  
MatrixILGPU target)
```

Parameters

source [MatrixILGPU](#)

Quellmatrix

value [double](#) ↗

Wert zum Abziehen

target [MatrixILGPU](#)

Zielmatrix

SynchronizeCPU()

Aus Geschwindigkeitsgründen muß manuell synchronisiert werden!

```
public void SynchronizeCPU()
```

SynchronizeCPU(MatrixILGPU)

Aus Geschwindigkeitsgründen muß manuell synchronisiert werden!

```
public static void SynchronizeCPU(MatrixILGPU matrix)
```

Parameters

matrix [MatrixILGPU](#)

zu behandelnde Daten

SynchronizeGPU()

Aus Geschwindigkeitsgründen muß manuell synchronisiert werden!

```
public void SynchronizeGPU()
```

SynchronizeGPU(MatrixILGPU)

Aus Geschwindigkeitsgründen muß manuell synchronisiert werden!

```
public static void SynchronizeGPU(MatrixILGPU matrix)
```

Parameters

matrix [MatrixILGPU](#)

zu behandelnde Daten

ToArray()

Schreibt den Inhalt der internen Matrix in ein Feld.

```
public double[] ToArray()
```

Returns

[double](#)[]

das Ergebnisfeld

ToArray(MatrixILGPU)

Schreibt den Inhalt der Matrix in ein Feld.

```
public static double[] ToArray(MatrixILGPU matrix)
```

Parameters

matrix [MatrixILGPU](#)

Returns

[double](#)[]

das Ergebnisfeld

ToArray(MatrixILGPU, double[])

Schreibt den Inhalt der Matrix in ein Feld.

```
public static void ToArray(MatrixILGPU matrix, double[] target)
```

Parameters

matrix [MatrixILGPU](#)

Quellmatrix

target [double](#)[]

Zielarray

ToSigmoid()

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: wendet die unsymmetrische logistische Funktion auf die interne Matrix an - verändert sie also.

```
public void ToSigmoid()
```

ToSigmoid(MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: liefert das Ergebnis der unsymmetrischen logistischen Funktion zurück.

```
public static MatrixILGPU ToSigmoid(MatrixILGPU source)
```

Parameters

source [MatrixILGPU](#)

die Eingabematrix

Returns

[MatrixILGPU](#)

die sigmoide Ergebnismatrix

ToSigmoid(MatrixILGPU, MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Inhaltlich: liefert das Ergebnis der unsymmetrischen logistischen Funktion im 'target' (beste Geschwindigkeit).

```
public static void ToSigmoid(MatrixILGPU source, MatrixILGPU target)
```

Parameters

source [MatrixILGPU](#)

die Eingabematrix

target [MatrixILGPU](#)

die Zielmatrix

ToSigmoid_instance_Kernel(Index2D, ArrayView2D<double, DenseX>)

ILGPU: Kernel für die Funktion

```
public static void ToSigmoid_instance_Kernel(Index2D index, ArrayView2D<double, Stride2D.DenseX> matrix)
```

Parameters

index Index2D

Position im 2D-Feld

matrix ArrayView2D<[double](#), Stride2D.DenseX>

die behandelten Daten

ToSigmoid_static_Kernel(Index2D, ArrayView2D<double, DenseX>, ArrayView2D<double, DenseX>)

ILGPU: Kernel für die Funktion

```
public static void ToSigmoid_static_Kernel(Index2D index, ArrayView2D<double, Stride2D.DenseX> source, ArrayView2D<double, Stride2D.DenseX> target)
```

Parameters

index Index2D

Position im 2D-Feld

source ArrayView2D<[double](#), Stride2D.DenseX>

Quellmatrix

target ArrayView2D<[double](#), Stride2D.DenseX>

Zielmatrix

ToString()

Standardausgabe der Matrix.

```
public override string ToString()
```

Returns

[string](#)

Stringrepräsentation der Matrix

Transpose(MatrixILGPU)

ILGPU: diese Funktion führt den dazugehörigen Kernel aus. Beachte bei 'static' muß die ganze Engine lokal bedient werden, aber die hier lebendigen Datenklassen können das schon alles haben.

Transponiert die Matrix.

```
public static MatrixILGPU Transpose(MatrixILGPU m)
```

Parameters

m [MatrixILGPU](#)

Eingabematrix

Returns

[MatrixILGPU](#)

Ergebnismatrix [m.sizeY, m.sizeX]

Transpose(MatrixILGPU, MatrixILGPU)

Transponiert die Eingabematrix in die Zielmatrix (beste Geschwindigkeit).

```
public static void Transpose(MatrixILGPU source, MatrixILGPU target)
```

Parameters

source [MatrixILGPU](#)

Eingabematrix

target [MatrixILGPU](#)

Zielmatrix

Transpose_static_Kernel(Index2D, ArrayView2D<double,
DenseX>, ArrayView2D<double, DenseX>)

ILGPU: Kernel für die Funktion

```
public static void Transpose_static_Kernel(Index2D index, ArrayView2D<double>,  
Stride2D.DenseX> source, ArrayView2D<double>, Stride2D.DenseX> target)
```

Parameters

index Index2D

"2D-Position in der Matrix

source ArrayView2D<[double](#)>, Stride2D.DenseX>

Quellmatrix

target ArrayView2D<[double](#)>, Stride2D.DenseX>

Zielmatrix

Class Pattern

Namespace: [MatrixFFN.Tools](#)

Assembly: MatrixFFN.dll

Diese Klasse soll Daten für die sigmoide Funktion normalisieren. Wertebereich in den abgebildet wird ist [0.25, 0.75] - so wird eine Fehlerkorrektur auch in den Extremen ermöglicht.

Diese Klasse kann für alle Daten genommen werden. Kategorisches sollte mit Pattern_Long verarbeitet werden.

```
public class Pattern
```

Inheritance

[object](#) ← Pattern

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Remarks

Bei Java hatte ich einen MathContext benutzt - hier gibt es 'decimal'. Dieser Datentyp soll präzise aber langsam sein. Meine Theorie war bestimmt gut und sollte auch zukunftsweisend sein. Wichtig ist die Abbildung auf das Intervall und zurück.

Deswegen werde ich alles mit 'decimal' realisieren - dachte ich. Dieser Wertetyp wird aber nicht von 'Math' unterstützt und ist eine Baustelle. So muß es mit 'double' gehen.

Constructors

Pattern()

Der parameterlose Konstruktor - für die Serialisierung.

```
public Pattern()
```

Pattern(double, double, double, double)

Der Konstruktor bekommt einen Wert aus dem Datenfeld und die Limits für das Normalisieren.

Möglich ist diese Schrittanzahl für den feineren Lernwert zu erhöhen.

```
public Pattern(double inWert, double inMin, double inMax, double inSchritte)
```

Parameters

inWert [double](#)

Wert aus dem Datenfeld

inMin [double](#)

kleinster Datenwert zum kodieren

inMax [double](#)

größter Datenwert zum kodieren

inSchritte [double](#)

Anzahl der Schritte im Feld

Pattern(BinaryReader)

Dieser Konstruktor lädt seine Daten aus dem BinaryReader und initialisiert sich damit.

```
public Pattern(BinaryReader reader)
```

Parameters

reader [BinaryReader](#)

ein BinaryReader

Fields

lernwert

der spezielle Alphawert = 1 / Schritte

```
public double lernwert
```

Field Value

double ↗

maxNorm

erwünschtes Spektrum in der sigmoiden Abbildung - 'norm'-Seite

```
public double maxNorm
```

Field Value

double ↗

maxReal

Die Limits des Werteverlaufs - die 'reale' Seite

```
public double maxReal
```

Field Value

double ↗

minNorm

erwünschtes Spektrum in der sigmoiden Abbildung - 'norm'-Seite

```
public double minNorm
```

Field Value

[double](#) ↗

minReal

Die Limits des Werteverlaufs - die 'reale' Seite

```
public double minReal
```

Field Value

[double](#) ↗

schritte

die Anzahl der Schritte zwischen den Extrema

```
public double schritte
```

Field Value

[double](#) ↗

schrittweiteNorm

Der kleinste Abstand zwischen den einzelnen Daten - 'norm'-Seite

```
public double schrittweiteNorm
```

Field Value

[double](#) ↗

schrittweiteReal

Der kleinste Abstand zwischen den einzelnen Daten - 'reale'-Seite

```
public double schrittweiteReal
```

Field Value

[double ↗](#)

wertNorm

lokaler Wert - 'norm'

```
public double wertNorm
```

Field Value

[double ↗](#)

wertReal

lokaler Wert - 'real'

```
public double wertReal
```

Field Value

[double ↗](#)

Methods

GetNormedValue(double)

Normalisieren eines Wertes.

```
public double GetNormedValue(double inWert)
```

Parameters

inWert [double](#)

zu normalisierender Wert

Returns

[double](#)

normalisierter Wert

GetRealValue(double)

Zückkonvertieren eines normalisierten Wertes in seine reale Wahrheit.

```
public double GetRealValue(double inWert)
```

Parameters

inWert [double](#)

normalisierter Wert zum konvertieren

Returns

[double](#)

der reale Wert

InfoString()

Liefert eine Übersicht über den Objektzustand und die eingestellten Limits.

```
public string InfoString()
```

Returns

[string](#)

der Infostring

Init(double, double, double, double)

Errechnet aus den Eckdaten die interne Normierungskodierung.

```
public void Init(double inWert, double inMin, double inMax, double inSchritte)
```

Parameters

inWert [double](#)

ein Wert aus dem Datenfeld

inMin [double](#)

unteres Wertelimit

inMax [double](#)

oberes Wertelimit

inSchritte [double](#)

die erwünschten Unterschritte zwischen den Limits.

LoadDataFromReader(BinaryReader)

eine traditionelle Laderoutine (binär)

```
public void LoadDataFromReader(BinaryReader reader)
```

Parameters

reader [BinaryReader](#)

ein BinaryReader

SaveDataToWriter(BinaryWriter)

eine traditionelle Speicherroutine (binär)

```
public void SaveDataToWriter(BinaryWriter writer)
```

Parameters

writer [BinaryWriter](#)

ein BinaryWriter

ToString()

Liefert eine Stringrepräsentation des Objektinhalts.

```
public override string ToString()
```

Returns

[string](#)

aktuellen Objektinhalt

TypeOf()

Identifiziert die Klasse.

```
public string TypeOf()
```

Returns

[string](#)

Klassenname

Class StopWatch

Namespace: [MatrixFFN.Tools](#)

Assembly: MatrixFFN.dll

Ein kleine Hilfsklasse, die die Ausgabe der 'Stopwatch' schön formatiert.

```
public class StopWatch
```

Inheritance

[object](#) ← StopWatch

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

StopWatch()

Standardkonstruktor

```
public StopWatch()
```

StopWatch(TimeSpan)

Diesem Konstruktor wird eine TimeSpan übergeben. Nützlich für das Speichern des Netzes.

```
public StopWatch(TimeSpan inputTimeSpan)
```

Parameters

[inputTimeSpan](#) [TimeSpan](#)

die abgesicherte 'localTimespan'

Fields

localTimespan

Die interne Zeitspanne der Stopuhr.

```
public TimeSpan localTimespan
```

Field Value

[TimeSpan ↗](#)

localWatch

Die interne Stopuhr.

```
public Stopwatch localWatch
```

Field Value

[Stopwatch ↗](#)

Methods

GetTimeSpan()

Liefert die TimeSpan - für Berechnungen besser geeignet

```
public TimeSpan GetTimeSpan()
```

Returns

[TimeSpan ↗](#)

die TimeSpan 'localTimespan'

Start()

Stopwatch.Start() wird ausgeführt.

```
public void Start()
```

Stop()

Stopwatch.Stop() wird ausgeführt.

```
public void Stop()
```

ToString()

Bei laufender StopUhr wird die Zeitspanne sich jeweils ändern. Ansonsten liefert diese Funktion die letzte abgeschlossene Zeitmessung.

```
public override string ToString()
```

Returns

[string](#)

die Zeitangabe