

UseCaseAnalyser.GraphVisualiser

Erzeugt von Doxygen 1.8.9.1

Die Jun 30 2015 08:46:23

Inhaltsverzeichnis

1	Verzeichnis der Namensbereiche	1
1.1	Pakete	1
2	Hierarchie-Verzeichnis	3
2.1	Klassenhierarchie	3
3	Klassen-Verzeichnis	5
3.1	Auflistung der Klassen	5
4	Datei-Verzeichnis	7
4.1	Auflistung der Dateien	7
5	Dokumentation der Namensbereiche	9
5.1	Paket UseCaseAnalyser	9
5.2	Paket UseCaseAnalyser.GraphVisualiser	9
5.3	Paket UseCaseAnalyser.GraphVisualiser.DrawingElements	9
6	Klassen-Dokumentation	11
6.1	UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine Klassenreferenz	11
6.1.1	Ausführliche Beschreibung	13
6.1.2	Dokumentation der Elementfunktionen	13
6.1.2.1	BeginCapChangedCallback	13
6.1.2.2	EndCapChangedCallback	13
6.1.2.3	LinePathChangedCallback	14
6.1.2.4	MeasureOverride	15
6.1.2.5	OnBeginCapChanged	15
6.1.2.6	OnEndCapChanged	15
6.1.2.7	OnLinePathChanged	16
6.1.2.8	OnRender	16
6.1.3	Dokumentation der Datenelemente	16
6.1.3.1	BeginCapProperty	16
6.1.3.2	EndCapProperty	17
6.1.3.3	LinePathProperty	17

6.1.3.4	StrokeProperty	17
6.1.3.5	StrokeThicknessProperty	17
6.1.4	Dokumentation der Propertys	17
6.1.4.1	BeginCap	17
6.1.4.2	EndCap	17
6.1.4.3	LinePath	17
6.1.4.4	Stroke	18
6.1.4.5	StrokeThickness	18
6.2	UseCaseAnalyser.GraphVisualiser.DrawingElements.ISelectableGraphElement Schnittstellenreferenz	18
6.2.1	Ausführliche Beschreibung	18
6.2.2	Dokumentation der Elementfunktionen	19
6.2.2.1	ChangeSelection	19
6.2.2.2	Select	19
6.3	UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge Klassenreferenz	19
6.3.1	Ausführliche Beschreibung	20
6.3.2	Dokumentation der Aufzählungstypen	20
6.3.2.1	DockedStatus	20
6.3.2.2	EdgeProcessType	20
6.3.3	Beschreibung der Konstruktoren und Destruktoren	20
6.3.3.1	UseCaseEdge	20
6.3.4	Dokumentation der Elementfunktionen	21
6.3.4.1	ChangeSelection	21
6.3.4.2	RecalcBezier	21
6.3.4.3	Select	21
6.3.4.4	SetDrawingBrush	21
6.3.4.5	Unselect	21
6.3.5	Dokumentation der Datenelemente	21
6.3.5.1	mDestUseCaseNode	21
6.3.5.2	mSourceUseCaseNode	21
6.3.5.3	mUnselectDrawingBrush	21
6.3.6	Dokumentation der Propertys	22
6.3.6.1	CurrentElement	22
6.3.6.2	DockPosDestElement	22
6.3.6.3	DockPosSourceElement	22
6.3.6.4	Edge	22
6.3.6.5	ProcessType	22
6.3.6.6	Selected	22
6.4	UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser Klassenreferenz	22
6.4.1	Ausführliche Beschreibung	24
6.4.2	Beschreibung der Konstruktoren und Destruktoren	24

6.4.2.1	UseCaseGraphVisualiser	24
6.4.3	Dokumentation der Elementfunktionen	24
6.4.3.1	AddEdge	24
6.4.3.2	AddNode	25
6.4.3.3	Background_OnPreviewMouseLeftButtonDown	26
6.4.3.4	CanvasScrollViewer_OnMouseWheel	26
6.4.3.5	Clear	26
6.4.3.6	GetPreviousNodeVariantCount	27
6.4.3.7	GraphVisualiser_OnMouseDown	28
6.4.3.8	GraphVisualiser_OnMouseMove	28
6.4.3.9	GraphVisualiser_OnMouseUp	28
6.4.3.10	RedrawGraph	29
6.4.3.11	ScenarioPropertyChanged	29
6.4.3.12	SetBrushForScenario	29
6.4.3.13	UseCasePropertyChanged	30
6.4.3.14	VisualiseEdges	30
6.4.3.15	VisualiseGraph	30
6.4.3.16	VisualiseNodes	31
6.4.4	Dokumentation der Datenelemente	31
6.4.4.1	CanvasSizeUpdateTime	31
6.4.4.2	ElementHeight	31
6.4.4.3	ElementWidth	31
6.4.4.4	GraphElementProperty	31
6.4.4.5	mLastSizeUpdateTime	31
6.4.4.6	mNodePosDict	31
6.4.4.7	mNodes	31
6.4.4.8	mOffsetElementPosition	31
6.4.4.9	mSelectedElement	31
6.4.4.10	ScaleRateZoom	31
6.4.4.11	ScenarioProperty	31
6.4.4.12	UseCaseProperty	31
6.4.5	Dokumentation der Property's	32
6.4.5.1	GraphElement	32
6.4.5.2	Scenario	32
6.4.5.3	UseCase	32
6.5	UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode Klassenreferenz	32
6.5.1	Ausführliche Beschreibung	33
6.5.2	Beschreibung der Konstruktoren und Destruktoren	33
6.5.2.1	UseCaseNode	33
6.5.3	Dokumentation der Elementfunktionen	33

6.5.3.1	AddEdge	33
6.5.3.2	ChangeSelection	33
6.5.3.3	GetCountOfEdges	34
6.5.3.4	GetEdgeIndex	35
6.5.3.5	RenderEdges	35
6.5.3.6	Select	35
6.5.3.7	SetDrawingBrush	35
6.5.3.8	Unselect	35
6.5.4	Dokumentation der Datenelemente	36
6.5.4.1	mEdges	36
6.5.4.2	mUnselectDrawingBrush	36
6.5.5	Dokumentation der Property's	36
6.5.5.1	CurrentElement	36
6.5.5.2	Node	36
6.5.5.3	Selected	36
7	Datei-Dokumentation	37
7.1	DrawingElements/CappedLine.cs-Dateireferenz	37
7.2	DrawingElements/ISelectableGraphElement.cs-Dateireferenz	37
7.3	DrawingElements/UseCaseEdge.xaml.cs-Dateireferenz	37
7.4	DrawingElements/UseCaseNode.xaml.cs-Dateireferenz	38
7.5	Properties/AssemblyInfo.cs-Dateireferenz	38
7.6	UseCaseGraphVisualiser.xaml.cs-Dateireferenz	38
Index		39

Kapitel 1

Verzeichnis der Namensbereiche

1.1 Pakete

Hier folgen die Pakete mit einer Kurzbeschreibung (wenn verfügbar):

UseCaseAnalyser	9
UseCaseAnalyser.GraphVisualiser	9
UseCaseAnalyser.GraphVisualiser.DrawingElements	9

Kapitel 2

Hierarchie-Verzeichnis

2.1 Klassenhierarchie

Die Liste der Ableitungen ist -mit Einschränkungen- alphabetisch sortiert:

FrameworkElement	
UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine	11
UseCaseAnalyser.GraphVisualiser.DrawingElements.ISelectableGraphElement	18
UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge	19
UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode	32
UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser	22

Kapitel 3

Klassen-Verzeichnis

3.1 Auflistung der Klassen

Hier folgt die Aufzählung aller Klassen, Strukturen, Varianten und Schnittstellen mit einer Kurzbeschreibung:

UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine	
Class for displaying a capped line. Sources from http://blogs.msdn.com/b/mrochon/archive/2011/01/asp.aspx	11
UseCaseAnalyser.GraphVisualiser.DrawingElements.ISelectableGraphElement	
An interface for classes that contain a GraphElement and should be selectable via the visualisation.	18
UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge	
Class for displaying a edge as a Bezier curve within a UseCaseGraph. On selection line color changes. It contains the edge to display as a reference.	19
UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser	
Class for displaying a UseCaseGraph element. It offers possibilities to select a single Graph↔Element while setting a dependency property GraphElement. Furthermore the user has the option to move all nodes.	22
UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode	
Class for displaying a node as rectangle within a UseCaseGraph. On selection border color changes. It contains the node to display as a reference.	32

Kapitel 4

Datei-Verzeichnis

4.1 Auflistung der Dateien

Hier folgt die Aufzählung aller Dateien mit einer Kurzbeschreibung:

UseCaseGraphVisualiser.xaml.cs	38
DrawingElements/ CappedLine.cs	37
DrawingElements/ ISelectableGraphElement.cs	37
DrawingElements/ UseCaseEdge.xaml.cs	37
DrawingElements/ UseCaseNode.xaml.cs	38
Properties/ AssemblyInfo.cs	38

Kapitel 5

Dokumentation der Namensbereiche

5.1 Paket UseCaseAnalyser

Namensbereiche

- package [GraphVisualiser](#)

5.2 Paket UseCaseAnalyser.GraphVisualiser

Namensbereiche

- package [DrawingElements](#)

Klassen

- class [UseCaseGraphVisualiser](#)

Class for displaying a UseCaseGraph element. It offers possibilities to select a single GraphElement while setting a dependency property GraphElement. Furthermore the user has the option to move all nodes.

5.3 Paket UseCaseAnalyser.GraphVisualiser.DrawingElements

Klassen

- class [CappedLine](#)

Class for displaying a capped line. Sources from <http://blogs.msdn.com/b/mrochon/archive/2011/01/10/custom-1.aspx>

- interface [ISelectableGraphElement](#)

An interface for classes that contain a GraphElement and should be selectable via the visualisation.

- class [UseCaseEdge](#)

Class for displaying a edge as a Bezier curve within a UseCaseGraph. On selection line color changes. It contains the edge to display as a reference.

- class [UseCaseNode](#)

Class for displaying a node as rectangle within a UseCaseGraph. On selection border color changes. It contains the node to display as a reference.

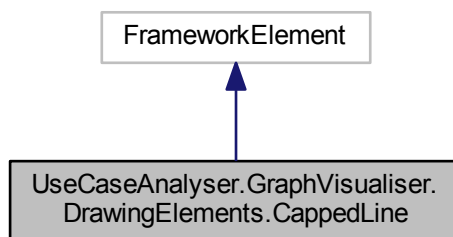
Kapitel 6

Klassen-Dokumentation

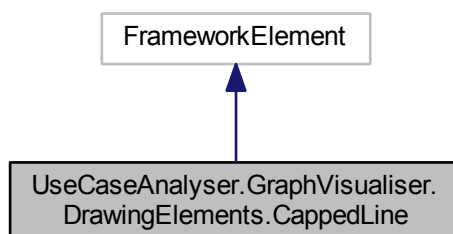
6.1 UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine Klassenreferenz

Class for displaying a capped line. Sources from <http://blogs.msdn.com/b/mrochon/archive/2011/01/10/custom.aspx>

Klassendiagramm für UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine:



Zusammengehörigkeiten von UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine:



Öffentliche Methoden

- virtual void [OnLinePathChanged](#) (PathGeometry value)
Property changed event handler for LinePath property.
- virtual void [OnBeginCapChanged](#) (Geometry value)
Property changed event handler for capped line property at the beginning of the LinePath.
- virtual void [OnEndCapChanged](#) (Geometry value)
Property changed event handler for capped line property at the end of the LinePath.

Statische öffentliche Attribute

- static readonly DependencyProperty [StrokeProperty](#) = Shape.StrokeProperty.AddOwner(typeof ([CappedLine](#)))
Binding configuration for a dependency property which is setting StrokeProperty
- static readonly DependencyProperty [StrokeThicknessProperty](#)
Binding configuration for a dependency property which is setting StrokeThicknessProperty
- static readonly DependencyProperty [LinePathProperty](#)
Binding configuration for a dependency property which is setting LinePathProperty
- static readonly DependencyProperty [BeginCapProperty](#)
Binding configuration for a dependency property which is setting BeginCapProperty
- static readonly DependencyProperty [EndCapProperty](#)
Binding configuration for a dependency property which is setting EndCapProperty

Geschützte Methoden

- override void [OnRender](#) (DrawingContext dc)
Logic for rendering a capped line
- override Size [MeasureOverride](#) (Size availableSize)
Overrides how size is measured

Propertys

- Brush [Stroke](#) [get, set]
Property for setting and getting Stroke
- double [StrokeThickness](#) [get, set]
Property for setting and getting StrokeThickness
- PathGeometry [LinePath](#) [get, set]
Property for setting and getting LinePath
- Geometry [BeginCap](#) [get, set]
Property for setting and getting cap at the start of the LinePath
- Geometry [EndCap](#) [get, set]
Property for setting and getting cap at the end of the LinePath

Private, statische Methoden

- static void [LinePathChangedCallback](#) (DependencyObject sender, DependencyPropertyChangedEventArgs args)
Callback handler used for LinePath dependency property
- static void [BeginCapChangedCallback](#) (DependencyObject sender, DependencyPropertyChangedEventArgs args)

Callback handler used for BeginCap dependency property

- static void **EndCapChangedCallback** (DependencyObject sender, DependencyPropertyChangedEventArgs args)

Callback handler used for EndCap dependency property

6.1.1 Ausführliche Beschreibung

Class for displaying a capped line. Sources from <http://blogs.msdn.com/b/mrochon/archive/2011/01/10/custom.aspx>

6.1.2 Dokumentation der Elementfunktionen

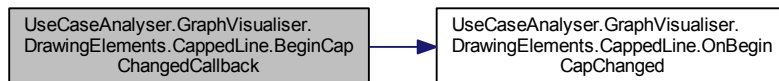
6.1.2.1 static void UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.BeginCapChangedCallback (DependencyObject sender, DependencyPropertyChangedEventArgs args) [static], [private]

Callback handler used for BeginCap dependency property

Parameter

<i>sender</i>	Dependency object that was changed
<i>args</i>	Event args containing information about the changes of the BeginCap property

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



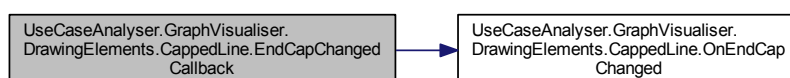
6.1.2.2 static void UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.EndCapChangedCallback (DependencyObject sender, DependencyPropertyChangedEventArgs args) [static], [private]

Callback handler used for EndCap dependency property

Parameter

<i>sender</i>	Dependency object that was changed
<i>args</i>	Event args containing information about the changes of the EndCap property

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



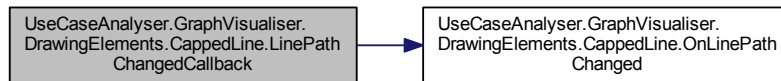
6.1.2.3 `static void UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.LinePathChangedCallback (DependencyObject sender, DependencyPropertyChangedEventArgs args)` `[static], [private]`

Callback handler used for LinePath dependency property

Parameter

<i>sender</i>	Dependency object that was changed
<i>args</i>	Event args containing information about the changes of the LinePath property

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



6.1.2.4 override Size UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.MeasureOverride (Size *availableSize*) [protected]

Overrides how size is measured

Parameter

<i>availableSize</i>	Size that is available
----------------------	------------------------

Rückgabe

New measured size

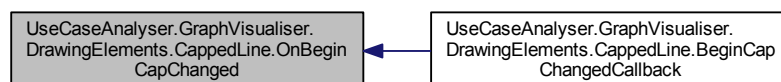
6.1.2.5 virtual void UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.OnBeginCapChanged (Geometry *value*) [virtual]

Property changed evented handler for capped line property at the beginning of the LinePath.

Parameter

<i>value</i>	new Geometry value
--------------	--------------------

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



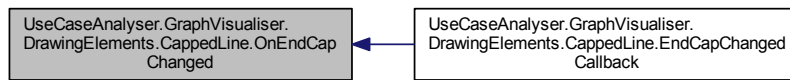
6.1.2.6 virtual void UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.OnEndCapChanged (Geometry *value*) [virtual]

Property changed evented handler for capped line property at the end of the LinePath.

Parameter

<i>value</i>	new Geometry value
--------------	--------------------

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



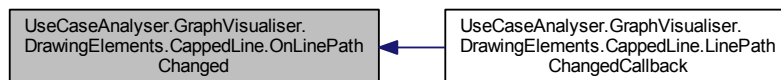
6.1.2.7 virtual void UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.OnLinePathChanged (PathGeometry *value*) [virtual]

Property changed evented handler for LinePath property.

Parameter

<i>value</i>	new PathGeometry value
--------------	------------------------

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



6.1.2.8 override void UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.OnRender (DrawingContext *dc*) [protected]

Logic for rendering a capped line

Parameter

<i>dc</i>	DrawingContext of the rendering capped line
-----------	---

6.1.3 Dokumentation der Datenelemente

6.1.3.1 readonly DependencyProperty UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.BeginCapProperty [static]

Initialisierung:

```

=
    DependencyProperty.Register("BeginCap", typeof (Geometry), typeof (CappedLine),
        new FrameworkPropertyMetadata(
            null,
            FrameworkPropertyMetadataOptions.AffectsRender,
            BeginCapChangedCallback))
  
```

Binding configuration for a dependency property which is setting BeginCapProperty

6.1.3.2 readonly DependencyProperty UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.EndCapProperty [static]

Initialisierung:

```
=
    DependencyProperty.Register("EndCap", typeof (Geometry), typeof (CappedLine),
        new FrameworkPropertyMetadata(
            null,
            FrameworkPropertyMetadataOptions.AffectsRender,
            EndCapChangedCallback))
```

Binding configuration for a dependency property which is setting EndCapProperty

6.1.3.3 readonly DependencyProperty UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.LinePathProperty [static]

Initialisierung:

```
=
    DependencyProperty.Register("LinePath", typeof (PathGeometry), typeof (CappedLine),
        new FrameworkPropertyMetadata(
            null,
            FrameworkPropertyMetadataOptions.AffectsRender,
            LinePathChangedCallback))
```

Binding configuration for a dependency property which is setting LinePathProperty

6.1.3.4 readonly DependencyProperty UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.StrokeProperty = Shape.StrokeProperty.AddOwner(typeof (CappedLine)) [static]

Binding configuration for a dependency property which is setting StrokeProperty

6.1.3.5 readonly DependencyProperty UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.StrokeThicknessProperty [static]

Initialisierung:

```
=
    Shape.StrokeThicknessProperty.AddOwner(typeof (CappedLine))
```

Binding configuration for a dependency property which is setting StrokeThicknessProperty

6.1.4 Dokumentation der Propertys

6.1.4.1 Geometry UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.BeginCap [get], [set]

Property for setting and getting cap at the start of the LinePath

6.1.4.2 Geometry UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.EndCap [get], [set]

Property for setting and getting cap at the end of the LinePath

6.1.4.3 PathGeometry UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.LinePath [get], [set]

Property for setting and getting LinePath

6.1.4.4 Brush UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.Stroke [get], [set]

Property for setting and getting Stroke

6.1.4.5 double UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine.StrokeThickness [get], [set]

Property for setting and getting StrokeThickness

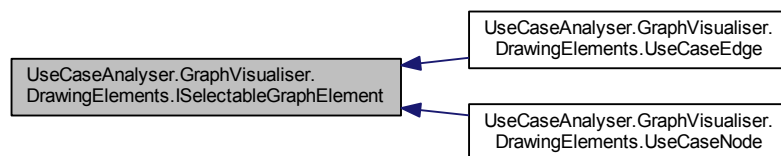
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- DrawingElements/[CappedLine.cs](#)

6.2 UseCaseAnalyser.GraphVisualiser.DrawingElements.ISelectableGraphElement Schnittstellenreferenz

An interface for classes that contain a GraphElement and should be selectable via the visualisation.

Klassendiagramm für UseCaseAnalyser.GraphVisualiser.DrawingElements.ISelectableGraphElement:



Öffentliche Methoden

- void [Select](#) ()
Change selection state to selected
- void [Unselect](#) ()
Reset selection state
- void [ChangeSelection](#) ()
Toggle selection state

Propertys

- bool [Selected](#) [get]
Check if object is selected
- IGraphElement [CurrentElement](#) [get]
Get reference to IGraphElement if selected

6.2.1 Ausführliche Beschreibung

An interface for classes that contain a GraphElement and should be selectable via the visualisation.

6.2.2 Dokumentation der Elementfunktionen

6.2.2.1 void UseCaseAnalyser.GraphVisualiser.DrawingElements.ISelectableGraphElement.ChangeSelection ()

Toggle selection state

Implementiert in [UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge](#) und [UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode](#).

6.2.2.2 void UseCaseAnalyser.GraphVisualiser.DrawingElements.ISelectableGraphElement.Select ()

Change selection state to selected

Implementiert in [UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge](#) und [UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode](#).

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:

6.3 UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge Klassenreferenz

Class for displaying a edge as a Bezier curve within a UseCaseGraph. On selection line color changes. It contains the edge to display as a reference.

Klassendiagramm für UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge:

Zusammengehörigkeiten von UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge:

Öffentliche Typen

- enum [DockedStatus](#) { [DockedStatus.Top](#), [DockedStatus.Bottom](#), [DockedStatus.Left](#), [DockedStatus.Right](#) }
Docked status of [CappedLine](#) on [UseCaseNode](#)
- enum [EdgeProcessType](#) { [EdgeProcessType.ForwardEdge](#), [EdgeProcessType.BackwardEdge](#) }
Type of [UseCaseEdge](#) which will be displayed

Öffentliche Methoden

- [UseCaseEdge](#) ([UseCaseNode](#) source, [UseCaseNode](#) dest, IEdge edge)
Creates a new instance of an visual presenter of an [UseCaseEdge](#)
- void [RecalcBezier](#) ()
Recalculation of the Bezier curve and redraw the Edge
- void [SetDrawingBrush](#) (Brush newBrush)
Set new brush color to this Edge
- void [Select](#) ()
Select this element
- void [Unselect](#) ()
Unselect this element
- void [ChangeSelection](#) ()
Switch selection status of this element

Öffentliche Attribute

- readonly [UseCaseNode](#) mDestUseCaseNode
Reference of destination visual [UseCaseNode](#)
- readonly [UseCaseNode](#) mSourceUseCaseNode
Reference of source visual [UseCaseNode](#)

Propertys

- IEdge [Edge](#) [get, private set]
Reference to the Edge in the UseCaseGraph
- bool [Selected](#) [get, private set]
Selected status of the element
- [DockedStatus](#) [DockPosSourceElement](#) [get, set]
Dock position of [CappedLine](#) on the source Element
- [DockedStatus](#) [DockPosDestElement](#) [get, set]
Dock Position Capped Line on the Destination Element
- [EdgeProcessType](#) [ProcessType](#) [get, set]
Process type of Edge which will be displayed
- IGraphElement [CurrentElement](#) [get]
Reference to the element in the UseCaseGraph

Private Attribute

- Brush [mUnselectDrawingBrush](#)
Brush which will be displayed if the element is not selected

6.3.1 Ausführliche Beschreibung

Class for displaying a edge as a Bezier curve within a UseCaseGraph. On selection line color changes. It contains the edge to display as a reference.

6.3.2 Dokumentation der Aufzählungstypen

6.3.2.1 enum UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge.DockedStatus

Docked status of [CappedLine](#) on [UseCaseNode](#)

Aufzählungswerte

- Top** [UseCaseEdge](#) is visualised on top of [UseCaseNode](#)
- Bottom** [UseCaseEdge](#) is visualised on bottom of [UseCaseNode](#)
- Left** [UseCaseEdge](#) is visualised on the left of [UseCaseNode](#)
- Right** [UseCaseEdge](#) is visualised on the right of [UseCaseNode](#)

6.3.2.2 enum UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge.EdgeProcessType

Type of [UseCaseEdge](#) which will be displayed

Aufzählungswerte

- ForwardEdge** [UseCaseEdge](#) is a forwarding edge
- BackwardEdge** [UseCaseEdge](#) is a backwarding edge ([UseCaseNode](#) with type [JumpNode](#) as source)

6.3.3 Beschreibung der Konstruktoren und Destruktoren

6.3.3.1 UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge.UseCaseEdge ([UseCaseNode](#) source, [UseCaseNode](#) dest, IEdge edge)

Creates a new instance of an visual presenter of an [UseCaseEdge](#)

Parameter

<i>source</i>	Source UseCaseNode
<i>dest</i>	Destination UseCaseNode
<i>edge</i>	Reference to the Edge in the Graph

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:

6.3.4 Dokumentation der Elementfunktionen

6.3.4.1 void UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge.ChangeSelection ()

Switch selection status of this element

Implementiert [UseCaseAnalyser.GraphVisualiser.DrawingElements.ISelectableGraphElement](#).

6.3.4.2 void UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge.RecalcBezier ()

Recalculation of the Bezier curve and redraw the Edge

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:

6.3.4.3 void UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge.Select ()

Select this element

Implementiert [UseCaseAnalyser.GraphVisualiser.DrawingElements.ISelectableGraphElement](#).

6.3.4.4 void UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge.SetDrawingBrush (Brush newBrush)

Set new brush color to this Edge

Parameter

<i>newBrush</i>	future color which will be used for drawing
-----------------	---

6.3.4.5 void UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge.Unselect ()

Unselect this element

Implementiert [UseCaseAnalyser.GraphVisualiser.DrawingElements.ISelectableGraphElement](#).

6.3.5 Dokumentation der Datenelemente

6.3.5.1 readonly UseCaseNode UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge.mDestUseCaseNode

Reference of destination visual [UseCaseNode](#)

6.3.5.2 readonly UseCaseNode UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge.mSourceUseCaseNode

Reference of source visual [UseCaseNode](#)

6.3.5.3 Brush UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge.mUnselectDrawingBrush [private]

Brush which will be displayed if the element is not selected

6.3.6 Dokumentation der Property

6.3.6.1 **IGraphElement** `UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge.CurrentElement` [get]

Reference to the element in the UseCaseGraph

6.3.6.2 **DockedStatus** `UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge.DockPosDestElement` [get], [set], [package]

Dock Position Capped Line on the Destination Element

6.3.6.3 **DockedStatus** `UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge.DockPosSourceElement` [get], [set], [package]

Dock position of [CappedLine](#) on the source Element

6.3.6.4 **IEdge** `UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge.Edge` [get], [private set]

Reference to the Edge in the UseCaseGraph

6.3.6.5 **EdgeProcessType** `UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge.ProcessType` [get], [set], [package]

Process type of Edge which will be displayed

6.3.6.6 **bool** `UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge.Selected` [get], [private set]

Selected status of the element

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- DrawingElements/[UseCaseEdge.xaml.cs](#)

6.4 UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser Klassenreferenz

Class for displaying a UseCaseGraph element. It offers possibilities to select a single GraphElement while setting a dependency property GraphElement. Furthermore the user has the option to move all nodes.

Öffentliche Methoden

- [UseCaseGraphVisualiser](#) ()
UseCaseGraphVisualiser default constructor
- void [RedrawGraph](#) ()
redraws the current usecasegraph -> nodes + edges are redrawn and the cache positon will be deleted

Statische öffentliche Attribute

- static readonly DependencyProperty [UseCaseProperty](#)
Binding configuration for a dependency property which is setting UseCaseGraph to display
- static readonly DependencyProperty [ScenarioProperty](#)

Binding configuration for a dependency property which is setting a scenario graph (which will be highlighted by UseCaseGraphVisualiser)

- static readonly DependencyProperty [GraphElementProperty](#)

Binding configuration for a dependency property which is setting the currently selected IGraphElement (INode/IEdge/IGraph) in UseCaseGraphVisualiser

Propertys

- IGraph [Scenario](#) [get, set]
Dependency property for currently selected scenario graph
- IGraphElement [GraphElement](#) [get, set]
Dependency property for currently selected IGraphElement
- UseCaseGraph [UseCase](#) [get, set]
Dependency property for use case graph that should be visualised

Private Methoden

- void [Clear](#) (bool clearCache=false)
Creates cache entries (current position) for all INode objects within the UseCaseNodes. Furthermore clear Canvas and mNodes list.
- void [VisualiseGraph](#) ()
Visualise all Nodes in Standard Position and redraw edges
- void [VisualiseNodes](#) ()
Visualise all nodes in dependency property UseCaseGraph.
- void [VisualiseEdges](#) ()
Visualise all edges by using Indices of start and end node to find corresponding UseCaseNodes. If UseCaseNodes are not contained in mNodes an InvalidOperationException will be thrown.
- void [AddNode](#) (INode node)
Adds a node to UseCaseGraphVisualiser canvas and node list. Furthermore adjusts default position of this node using NormalIndex/VariantIndex attributes if no cached value is given.
- int [GetPreviousNodeVariantCount](#) (UseCaseNode ucNode)
Calculates additional offset corresponding to previous node's variant sequence count.
- void [SetBrushForScenario](#) (IGraph sourceGraph, Brush futureBrush)
Color for specific Scenario will be set
- void [AddEdge](#) (UseCaseNode startNode, UseCaseNode endingNode, IEdge edge)
Adds an edge to Canvas and triggers start and ending node to render their edges.
- void [Background_OnPreviewMouseLeftButtonDown](#) (object sender, MouseButtonEventArgs e)
Event handler for canvas. Unselect all selectable elements in canvas and set dependency property GraphElement to UseCase.
- void [GraphVisualiser_OnMouseDown](#) (object sender, MouseButtonEventArgs e)
Event handler for Graphvisualiser. Determines if a selectable UseCaseNode/Edge was pressed and updates GraphElement.
- void [GraphVisualiser_OnMouseMove](#) (object sender, MouseEventArgs e)
Event handler for Graphvisualiser. Handles dragging state.
- void [GraphVisualiser_OnMouseUp](#) (object sender, MouseButtonEventArgs e)
Event handler for Graphvisualiser. Resets dragging state.
- void [CanvasScrollViewer_OnMouseWheel](#) (object sender, MouseWheelEventArgs e)
Event handler for CanvasScrollViewer. If left Ctrl Key is pressed Canvas Zoom starts

Private, statische Methoden

- static void [ScenarioPropertyChanged](#) (DependencyObject d, DependencyPropertyChangedEventArgs e)
Property changed evented handler for Scenerio property. If Scenario property is modified [UseCaseGraphVisualiser](#) will highlight them within the view.
- static void [UseCasePropertyChanged](#) (DependencyObject d, DependencyPropertyChangedEventArgs e)
Property changed evented handler for UseCase property. If UseCase property is modified [UseCaseGraphVisualiser](#) will visualise the new UseCaseGraph. Furthermore if this UseCaseGraph was already displayed the cached positions are used instead of calculating them again from scratch.

Private Attribute

- const double [ElementWidth](#) = 110
- const double [ElementHeight](#) = 70
- const double [ScaleRateZoom](#) = 1.05
- readonly List< [UseCaseNode](#) > [mNodes](#) = new List<[UseCaseNode](#)>()
- readonly Dictionary< INode, Point > [mNodePosDict](#) = new Dictionary<INode, Point>()
- Point [mOffsetElementPosition](#)
- FrameworkElement [mSelectedElement](#)
- DateTime [mLastSizeUpdateTime](#) = DateTime.Now
- const uint [CanvasSizeUpdateTime](#) = 20

6.4.1 Ausführliche Beschreibung

Class for displaying a UseCaseGraph element. It offers possibilities to select a single GraphElement while setting a dependency property GraphElement. Furthermore the user has the option to move all nodes.

6.4.2 Beschreibung der Konstruktoren und Destruktoren

6.4.2.1 UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.UseCaseGraphVisualiser ()

[UseCaseGraphVisualiser](#) default constructor

6.4.3 Dokumentation der Elementfunktionen

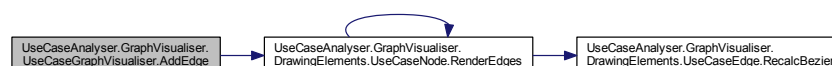
6.4.3.1 void UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.AddEdge ([UseCaseNode](#) *startNode*, [UseCaseNode](#) *endingNode*, [IEdge](#) *edge*) [private]

Adds an edge to Canvas and triggers start and ending node to render their edges.

Parameter

<i>startNode</i>	Node where edge starts.
<i>endingNode</i>	Node where edge ends.
<i>edge</i>	IEdge which will be wrapped within an UseCaseEdge.

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



6.4.3.2 void UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.AddNode (INode *node*) [private]

Adds a node to [UseCaseGraphVisualiser](#) canvas and node list. Furthermore adjusts default position of this node using NormalIndex/VariantIndex attributes if no cached value is given.

Parameter

<i>node</i>	INode object that should be wrapped within a UseCaseNode.
-------------	---

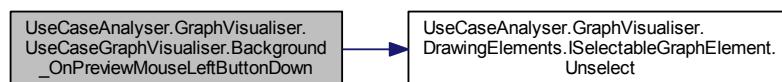
6.4.3.3 void UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.Background_OnPreviewMouseLeftButtonDown (object *sender*, MouseEventArgs *e*) [private]

Event handler for canvas. Unselect all selectable elements in canvas and set dependency property GraphElement to UseCase.

Parameter

<i>sender</i>	Sender of Background_OnPreviewMouseLeftButtonDown event.
<i>e</i>	Background_OnPreviewMouseLeftButtonDown mouse button event arguments.

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



6.4.3.4 void UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.CanvasScrollViewer_OnMouseWheel (object *sender*, MouseEventArgs *e*) [private]

Event handler for CanvasScrollViewer. If left Ctrl Key is pressed Canvas Zoom starts

Parameter

<i>sender</i>	Sender of CanvasScrollViewer_OnMouseWheel event
<i>e</i>	CanvasScrollViewer_OnMouseWheel mouse wheel event arguments

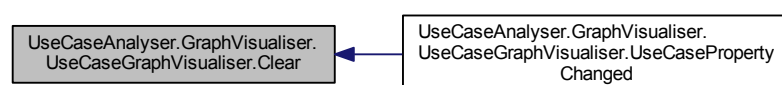
6.4.3.5 void UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.Clear (bool *clearCache* = false) [private]

Creates cache entries (current position) for all INode objects within the UseCaseNodes. Furthermore clear Canvas and mNodes list.

Parameter

<i>clearCache</i>	True:clear cache of nodes False: save old postion of nodes
-------------------	--

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



6.4.3.6 `int UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.GetPreviousNodeVariantCount (UseCaseNode ucNode) [private]`

Calculates additional offset corresponding to previous node's variant sequence count.

Parameter

<i>ucNode</i>	UseCaseNode which predecessor's variant sequence count should be determined.
---------------	--

Rückgabe

Count of previous normal node - if ucNode is not a normal node itself or previous could not be determined count is 0.

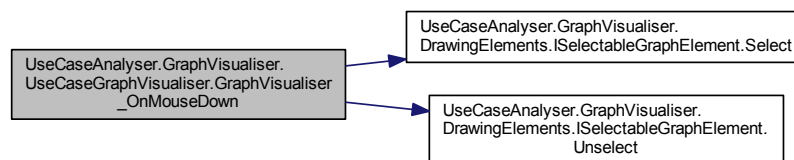
6.4.3.7 `void UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.GraphVisualiser_OnMouseDown (object sender, MouseButtonEventArgs e) [private]`

Event handler for Graphvisualiser. Determines if a selectable UseCaseNode/Egde was pressed and updates GraphElement.

Parameter

<i>sender</i>	Sender of GraphVisualiser_OnMouseDown event.
<i>e</i>	GraphVisualiser_OnMouseDown mouse button event arguments.

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



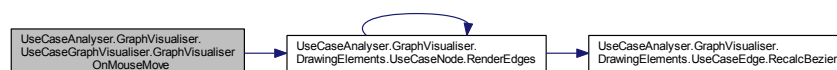
6.4.3.8 `void UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.GraphVisualiser_OnMouseMove (object sender, MouseEventArgs e) [private]`

Event handler for Graphvisualiser. Handles dragging state.

Parameter

<i>sender</i>	Sender of GraphVisualiser_OnMouseMove event.
<i>e</i>	GraphVisualiser_OnMouseMove mouse button event arguments.

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



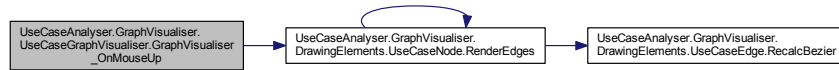
6.4.3.9 `void UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.GraphVisualiser_OnMouseUp (object sender, MouseButtonEventArgs e) [private]`

Event handler for Graphvisualiser. Resets dragging state.

Parameter

<i>sender</i>	Sender of GraphVisualiser_OnMouseUp event.
<i>e</i>	GraphVisualiser_OnMouseUp mouse button event arguments.

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



6.4.3.10 void UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.RedrawGraph ()

redraws the current usecasegraph → nodes + edges are redrawn and the cache position will be deleted

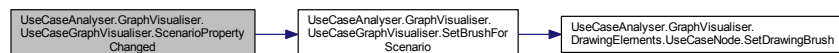
6.4.3.11 static void UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.ScenarioPropertyChanged (DependencyObject *d*, DependencyPropertyChangedEventArgs *e*) [static], [private]

Property changed event handler for Scenario property. If Scenario property is modified [UseCaseGraphVisualiser](#) will highlight them within the view.

Parameter

<i>d</i>	Dependency object that was changed
<i>e</i>	Event args containing information about the changes of the Scenario property

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:

6.4.3.12 void UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.SetBrushForScenario (IGraph *sourceGraph*, Brush *futureBrush*) [private]

Color for specific Scenario will be set

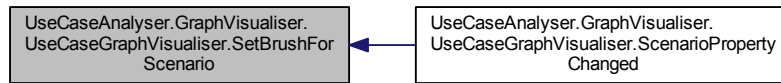
Parameter

<i>sourceGraph</i>	Scenario which will be highlighted
<i>futureBrush</i>	Brush which will be used to highlight the specific scenario

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



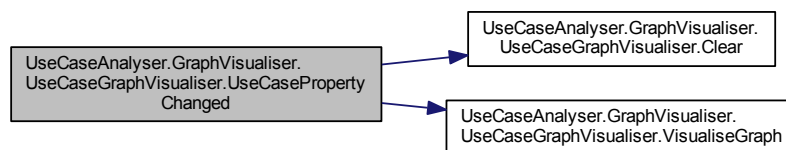
6.4.3.13 `static void UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.UseCasePropertyChanged (DependencyObject d, DependencyPropertyChangedEventArgs e) [static], [private]`

Property changed event handler for UseCase property. If UseCase property is modified [UseCaseGraphVisualiser](#) will visualise the new UseCaseGraph. Furthermore if this UseCaseGraph was already displayed the cached positions are used instead of calculating them again from scratch.

Parameter

<i>d</i>	Dependency object that was changed
<i>e</i>	Event args containing information about the changes of the UseCase property

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:



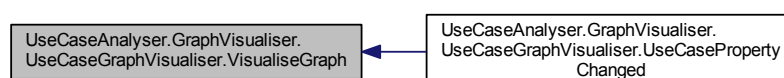
6.4.3.14 `void UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.VisualiseEdges () [private]`

Visualise all edges by using Indices of start and end node to find corresponding UseCaseNodes. If UseCaseNodes are not contained in mNodes an InvalidOperationException will be thrown.

6.4.3.15 `void UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.VisualiseGraph () [private]`

Visualise all Nodes in Standard Position and redraw edges

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:



6.4.3.16 void UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.VisualiseNodes () [private]

Visualise all nodes in dependency property UseCaseGraph.

6.4.4 Dokumentation der Datenelemente

6.4.4.1 const uint UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.CanvasSizeUpdateTime = 20 [private]

6.4.4.2 const double UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.ElementHeight = 70 [private]

6.4.4.3 const double UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.ElementWidth = 110 [private]

6.4.4.4 readonly DependencyProperty UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.GraphElementProperty
[static]

Initialisierung:

```
= DependencyProperty.Register("GraphElement",
    typeof (IGraphElement), typeof (UseCaseGraphVisualiser))
```

Binding configuration for a dependency property which is setting the currently selected IGraphElement (INode/I↔Edge/IGraph) in UseCaseGraphVisualiser

6.4.4.5 DateTime UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.mLastSizeUpdateTime = DateTime.Now
[private]

6.4.4.6 readonly Dictionary<INode, Point> UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.mNodePosDict = new
Dictionary<INode, Point>() [private]

6.4.4.7 readonly List<UseCaseNode> UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.mNodes = new
List<UseCaseNode>() [private]

6.4.4.8 Point UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.mOffsetElementPosition [private]

6.4.4.9 FrameworkElement UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.mSelectedElement [private]

6.4.4.10 const double UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.ScaleRateZoom = 1.05 [private]

6.4.4.11 readonly DependencyProperty UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.ScenarioProperty
[static]

Initialisierung:

```
= DependencyProperty.Register("Scenario",
    typeof (IGraph), typeof (UseCaseGraphVisualiser), new PropertyMetadata(
    ScenarioPropertyChanged))
```

Binding configuration for a dependency property which is setting a scenario graph (which will be highlighted by UseCaseGraphVisualiser)

6.4.4.12 readonly DependencyProperty UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.UseCaseProperty
[static]

Initialisierung:

```
= DependencyProperty.Register("UseCase",
    typeof(UseCaseGraph), typeof(UseCaseGraphVisualiser), new
    PropertyMetadata(UseCasePropertyChanged))
```

Binding configuration for a dependency property which is setting UseCaseGraph to display

6.4.5 Dokumentation der Propertys

6.4.5.1 IGraphElement UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.GraphElement [get], [set]

Dependency property for currently selected IGraphElement

6.4.5.2 IGraph UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.Scenario [get], [set]

Dependency property for currently selected scenario graph

6.4.5.3 UseCaseGraph UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser.UseCase [get], [set]

Dependency property for use case graph that should be visualised

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [UseCaseGraphVisualiser.xaml.cs](#)

6.5 UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode Klassenreferenz

Class for displaying a node as rectangle within a UseCaseGraph. On selection border color changes. It contains the node to display as a reference.

Klassendiagramm für UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode:

Zusammengehörigkeiten von UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode:

Öffentliche Methoden

- [UseCaseNode](#) (INode node)
Wrapper class for GraphFramework's INode which is used to define how a node will be displayed in [UseCaseGraphVisualiser](#).
- void [RenderEdges](#) ([UseCaseNode](#) notRenderNode=null)
Recursive function for rendering edges of this node by using RecalcBezier and its neighbours.
- void [AddEdge](#) ([UseCaseEdge](#) newEdge)
Add an edge to [UseCaseNode](#) if not already contained and nodes is either starting or endpoint of the specified edge.
- int [GetEdgeIndex](#) ([UseCaseEdge](#) sourceEdge)
Return index of specified edge corresponding to its DockedStatus.
- int [GetCountOfEdges](#) ([UseCaseEdge](#) sourceEdge)
Counts the amount of edges in depending of the docking status
- void [SetDrawingBrush](#) (IEnumerable< IEdge > toColorEdges, Brush newBrush)
Color for specific scenario will be set
- void [Select](#) ()
Select this element (visualised by glowing)
- void [Unselect](#) ()
Unselect this element (remove glow effect)
- void [ChangeSelection](#) ()
Switch selection status of this element

Öffentliche Attribute

- readonly List< [UseCaseEdge](#) > [mEdges](#) = new List<[UseCaseEdge](#)>()
List of [UseCaseEdge](#) that either start or end in this [UseCaseNode](#)

Propertys

- INode [Node](#) [get, private set]
Node property for GraphFrameworks INode element which is wrapped by this class.
- bool [Selected](#) [get, private set]
Property to check if [UseCaseNode](#) is marked as selected
- IGraphElement [CurrentElement](#) [get]
Reference to the element in the Graph

Private Attribute

- Brush [mUnselectDrawingBrush](#)
Brush which will be displayed if the element is not selected

6.5.1 Ausführliche Beschreibung

Class for displaying a node as rectangle within a UseCaseGraph. On selection border color changes. It contains the node to display as a reference.

6.5.2 Beschreibung der Konstruktoren und Destruktoren

6.5.2.1 UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode.UseCaseNode (INode *node*)

Wrapper class for GraphFrameworks's INode which is used to define how a node will be displayed in [UseCaseGraphVisualiser](#).

Parameter

<i>node</i>	INode object that will be wrapped by UseCaseNode
-------------	--

6.5.3 Dokumentation der Elementfunktionen

6.5.3.1 void UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode.AddEdge ([UseCaseEdge](#) *newEdge*)

Add an edge to [UseCaseNode](#) if not already contained and nodes is either starting or endpoint of the specified edge.

Parameter

<i>newEdge</i>	UseCaseEdge that should be added to the UseCaseNode
----------------	---

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:

6.5.3.2 void UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode.ChangeSelection ()

Switch selection status of this element

Implementiert [UseCaseAnalyser.GraphVisualiser.DrawingElements.ISelectableGraphElement](#).

6.5.3.3 `int UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode.GetCountOfEdges (UseCaseEdge
sourceEdge)`

Counts the amount of edges in depending of the docking status

Parameter

<i>sourceEdge</i>	Elements will be counted by the position of this element
-------------------	--

Rückgabe

amount of Edges at the same docking status of this node

6.5.3.4 `int UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode.GetEdgeIndex (UseCaseEdge sourceEdge)`

Return index of specified edge corresponding to its DockedStatus.

Parameter

<i>sourceEdge</i>	Edge for determine index.
-------------------	---------------------------

Rückgabe

Number of index.

6.5.3.5 `void UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode.RenderEdges (UseCaseNode notRenderNode = null)`

Recursive function for rendering edges of this node by using RecalcBezier and its neighbours.

Parameter

<i>notRenderNode</i>	Optional parameter for use case node which prevents rendering of specified node's edges.
----------------------	--

Hier ist ein Graph, der zeigt, was diese Funktion aufruft:

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:

6.5.3.6 `void UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode.Select ()`

Select this element (visualised by glowing)

Implementiert [UseCaseAnalyser.GraphVisualiser.DrawingElements.ISelectableGraphElement](#).

6.5.3.7 `void UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode.SetDrawingBrush (IEnumerable< IEdge > toColorEdges, Brush newBrush)`

Color for specific scenario will be set

Parameter

<i>toColorEdges</i>	List of Edges which will be colored
<i>newBrush</i>	Brush which will be used to highlite the specific scenario

Hier ist ein Graph der zeigt, wo diese Funktion aufgerufen wird:

6.5.3.8 `void UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode.Unselect ()`

Unselect this element (remove glow effect)

Implementiert [UseCaseAnalyser.GraphVisualiser.DrawingElements.ISelectableGraphElement](#).

6.5.4 Dokumentation der Datenelemente

6.5.4.1 `readonly List<UseCaseEdge> UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode.mEdges = new List<UseCaseEdge>()`

List of [UseCaseEdge](#) that either start or end in this [UseCaseNode](#)

6.5.4.2 `Brush UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode.mUnselectDrawingBrush [private]`

Brush which will be displayed if the element is not selected

6.5.5 Dokumentation der Propertys

6.5.5.1 `IGraphElement UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode.CurrentElement [get]`

Reference to the element in the Graph

6.5.5.2 `INode UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode.Node [get],[private set]`

Node property for GraphFrameworks INode element which is wrapped by this class.

6.5.5.3 `bool UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode.Selected [get],[private set]`

Property to check if [UseCaseNode](#) is marked as selected

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [DrawingElements/UseCaseNode.xaml.cs](#)

Kapitel 7

Datei-Dokumentation

7.1 DrawingElements/CappedLine.cs-Dateireferenz

Klassen

- class [UseCaseAnalyser.GraphVisualiser.DrawingElements.CappedLine](#)

Class for displaying a capped line. Sources from <http://blogs.msdn.com/b/mrochon/archive/2011/01/10/custom-l.aspx>

Namensbereiche

- package [UseCaseAnalyser.GraphVisualiser.DrawingElements](#)

7.2 DrawingElements/ISelectableGraphElement.cs-Dateireferenz

Klassen

- interface [UseCaseAnalyser.GraphVisualiser.DrawingElements.ISelectableGraphElement](#)

An interface for classes that contain a GraphElement and should be selectable via the visualisation.

Namensbereiche

- package [UseCaseAnalyser.GraphVisualiser.DrawingElements](#)

7.3 DrawingElements/UseCaseEdge.xaml.cs-Dateireferenz

Klassen

- class [UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseEdge](#)

Class for displaying a edge as a Bezier curve within a UseCaseGraph. On selection line color changes. It contains the edge to display as a reference.

Namensbereiche

- package [UseCaseAnalyser.GraphVisualiser.DrawingElements](#)

7.4 DrawingElements/UseCaseNode.xaml.cs-Dateireferenz

Klassen

- class [UseCaseAnalyser.GraphVisualiser.DrawingElements.UseCaseNode](#)
Class for displaying a node as rectangle within a UseCaseGraph. On selection border color changes. It contains the node to display as a reference.

Namensbereiche

- package [UseCaseAnalyser.GraphVisualiser.DrawingElements](#)

7.5 Properties/AssemblyInfo.cs-Dateireferenz

7.6 UseCaseGraphVisualiser.xaml.cs-Dateireferenz

Klassen

- class [UseCaseAnalyser.GraphVisualiser.UseCaseGraphVisualiser](#)
Class for displaying a UseCaseGraph element. It offers possibilities to select a single GraphElement while setting a dependency property GraphElement. Furthermore the user has the option to move all nodes.

Namensbereiche

- package [UseCaseAnalyser.GraphVisualiser](#)

Index

- AddEdge
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::UseCaseNode, [33](#)
 - UseCaseAnalyser::GraphVisualiser::UseCase↔
GraphVisualiser, [24](#)
- AddNode
 - UseCaseAnalyser::GraphVisualiser::UseCase↔
GraphVisualiser, [24](#)
- Background_OnPreviewMouseLeftButtonDown
 - UseCaseAnalyser::GraphVisualiser::UseCase↔
GraphVisualiser, [26](#)
- BackwardEdge
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::UseCaseEdge, [20](#)
- BeginCap
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::CappedLine, [17](#)
- BeginCapChangedCallback
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::CappedLine, [13](#)
- BeginCapProperty
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::CappedLine, [16](#)
- Bottom
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::UseCaseEdge, [20](#)
- CanvasScrollViewer_OnMouseWheel
 - UseCaseAnalyser::GraphVisualiser::UseCase↔
GraphVisualiser, [26](#)
- CanvasSizeUpdateTime
 - UseCaseAnalyser::GraphVisualiser::UseCase↔
GraphVisualiser, [31](#)
- ChangeSelection
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::ISelectableGraphElement, [19](#)
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::UseCaseEdge, [21](#)
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::UseCaseNode, [33](#)
- Clear
 - UseCaseAnalyser::GraphVisualiser::UseCase↔
GraphVisualiser, [26](#)
- CurrentElement
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::UseCaseEdge, [22](#)
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::UseCaseNode, [36](#)
- DockPosDestElement
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::UseCaseEdge, [22](#)
- DockPosSourceElement
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::UseCaseEdge, [22](#)
- DockedStatus
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::UseCaseEdge, [20](#)
- DrawingElements/CappedLine.cs, [37](#)
- DrawingElements/ISelectableGraphElement.cs, [37](#)
- DrawingElements/UseCaseEdge.xaml.cs, [37](#)
- DrawingElements/UseCaseNode.xaml.cs, [38](#)
- Edge
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::UseCaseEdge, [22](#)
- EdgeProcessType
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::UseCaseEdge, [20](#)
- ElementHeight
 - UseCaseAnalyser::GraphVisualiser::UseCase↔
GraphVisualiser, [31](#)
- ElementWidth
 - UseCaseAnalyser::GraphVisualiser::UseCase↔
GraphVisualiser, [31](#)
- EndCap
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::CappedLine, [17](#)
- EndCapChangedCallback
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::CappedLine, [13](#)
- EndCapProperty
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::CappedLine, [16](#)
- ForwardEdge
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::UseCaseEdge, [20](#)
- GetCountOfEdges
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::UseCaseNode, [33](#)
- GetEdgeIndex
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
Elements::UseCaseNode, [35](#)
- GetPreviousNodeVariantCount
 - UseCaseAnalyser::GraphVisualiser::UseCase↔
GraphVisualiser, [26](#)
- GraphElement

UseCaseAnalyser::GraphVisualiser::UseCase↔
 GraphVisualiser, [32](#)
 GraphElementProperty
 UseCaseAnalyser::GraphVisualiser::UseCase↔
 GraphVisualiser, [31](#)
 GraphVisualiser_OnMouseDown
 UseCaseAnalyser::GraphVisualiser::UseCase↔
 GraphVisualiser, [28](#)
 GraphVisualiser_OnMouseMove
 UseCaseAnalyser::GraphVisualiser::UseCase↔
 GraphVisualiser, [28](#)
 GraphVisualiser_OnMouseUp
 UseCaseAnalyser::GraphVisualiser::UseCase↔
 GraphVisualiser, [28](#)
 Left
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::UseCaseEdge, [20](#)
 LinePath
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::CappedLine, [17](#)
 LinePathChangedCallback
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::CappedLine, [13](#)
 LinePathProperty
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::CappedLine, [17](#)
 mDestUseCaseNode
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::UseCaseEdge, [21](#)
 mEdges
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::UseCaseNode, [36](#)
 mLastSizeUpdateTime
 UseCaseAnalyser::GraphVisualiser::UseCase↔
 GraphVisualiser, [31](#)
 mNodePosDict
 UseCaseAnalyser::GraphVisualiser::UseCase↔
 GraphVisualiser, [31](#)
 mNodes
 UseCaseAnalyser::GraphVisualiser::UseCase↔
 GraphVisualiser, [31](#)
 mOffsetElementPosition
 UseCaseAnalyser::GraphVisualiser::UseCase↔
 GraphVisualiser, [31](#)
 mSelectedElement
 UseCaseAnalyser::GraphVisualiser::UseCase↔
 GraphVisualiser, [31](#)
 mSourceUseCaseNode
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::UseCaseEdge, [21](#)
 mUnselectDrawingBrush
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::UseCaseEdge, [21](#)
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::UseCaseNode, [36](#)
 MeasureOverride
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::CappedLine, [15](#)
 Node
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::UseCaseNode, [36](#)
 OnBeginCapChanged
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::CappedLine, [15](#)
 OnEndCapChanged
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::CappedLine, [15](#)
 OnLinePathChanged
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::CappedLine, [16](#)
 OnRender
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::CappedLine, [16](#)
 ProcessType
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::UseCaseEdge, [22](#)
 Properties/AssemblyInfo.cs, [38](#)
 RecalcBezier
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::UseCaseEdge, [21](#)
 RedrawGraph
 UseCaseAnalyser::GraphVisualiser::UseCase↔
 GraphVisualiser, [29](#)
 RenderEdges
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::UseCaseNode, [35](#)
 Right
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::UseCaseEdge, [20](#)
 ScaleRateZoom
 UseCaseAnalyser::GraphVisualiser::UseCase↔
 GraphVisualiser, [31](#)
 Scenario
 UseCaseAnalyser::GraphVisualiser::UseCase↔
 GraphVisualiser, [32](#)
 ScenarioProperty
 UseCaseAnalyser::GraphVisualiser::UseCase↔
 GraphVisualiser, [31](#)
 ScenarioPropertyChanged
 UseCaseAnalyser::GraphVisualiser::UseCase↔
 GraphVisualiser, [29](#)
 Select
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::ISelectableGraphElement, [19](#)
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::UseCaseEdge, [21](#)
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::UseCaseNode, [35](#)
 Selected
 UseCaseAnalyser::GraphVisualiser::Drawing↔
 Elements::UseCaseEdge, [22](#)

- UseCaseAnalyser::GraphVisualiser::Drawing↔
 - Elements::UseCaseNode, 36
- SetBrushForScenario
 - UseCaseAnalyser::GraphVisualiser::UseCase↔
 - GraphVisualiser, 29
- SetDrawingBrush
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
 - Elements::UseCaseEdge, 21
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
 - Elements::UseCaseNode, 35
- Stroke
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
 - Elements::CappedLine, 17
- StrokeProperty
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
 - Elements::CappedLine, 17
- StrokeThickness
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
 - Elements::CappedLine, 18
- StrokeThicknessProperty
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
 - Elements::CappedLine, 17
- Top
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
 - Elements::UseCaseEdge, 20
- Unselect
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
 - Elements::UseCaseEdge, 21
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
 - Elements::UseCaseNode, 35
- UseCase
 - UseCaseAnalyser::GraphVisualiser::UseCase↔
 - GraphVisualiser, 32
- UseCaseAnalyser, 9
- UseCaseAnalyser.GraphVisualiser, 9
- UseCaseAnalyser.GraphVisualiser.DrawingElements, 9
- UseCaseAnalyser.GraphVisualiser.DrawingElements↔
 - CappedLine, 11
- UseCaseAnalyser.GraphVisualiser.DrawingElements↔
 - ISelectableGraphElement, 18
- UseCaseAnalyser.GraphVisualiser.DrawingElements↔
 - UseCaseEdge, 19
- UseCaseAnalyser.GraphVisualiser.DrawingElements↔
 - UseCaseNode, 32
- UseCaseAnalyser.GraphVisualiser.UseCaseGraph↔
 - Visualiser, 22
- UseCaseAnalyser::GraphVisualiser::DrawingElements↔
 - ::CappedLine
 - BeginCap, 17
 - BeginCapChangedCallback, 13
 - BeginCapProperty, 16
 - EndCap, 17
 - EndCapChangedCallback, 13
 - EndCapProperty, 16
 - LinePath, 17
 - LinePathChangedCallback, 13
 - LinePathProperty, 17
 - MeasureOverride, 15
 - OnBeginCapChanged, 15
 - OnEndCapChanged, 15
 - OnLinePathChanged, 16
 - OnRender, 16
 - Stroke, 17
 - StrokeProperty, 17
 - StrokeThickness, 18
 - StrokeThicknessProperty, 17
- UseCaseAnalyser::GraphVisualiser::DrawingElements↔
 - ::ISelectableGraphElement
 - ChangeSelection, 19
 - Select, 19
- UseCaseAnalyser::GraphVisualiser::DrawingElements↔
 - ::UseCaseEdge
 - BackwardEdge, 20
 - Bottom, 20
 - ChangeSelection, 21
 - CurrentElement, 22
 - DockPosDestElement, 22
 - DockPosSourceElement, 22
 - DockedStatus, 20
 - Edge, 22
 - EdgeProcessType, 20
 - ForwardEdge, 20
 - Left, 20
 - mDestUseCaseNode, 21
 - mSourceUseCaseNode, 21
 - mUnselectDrawingBrush, 21
 - ProcessType, 22
 - RecalcBezier, 21
 - Right, 20
 - Select, 21
 - Selected, 22
 - SetDrawingBrush, 21
 - Top, 20
 - Unselect, 21
 - UseCaseEdge, 20
- UseCaseAnalyser::GraphVisualiser::DrawingElements↔
 - ::UseCaseNode
 - AddEdge, 33
 - ChangeSelection, 33
 - CurrentElement, 36
 - GetCountOfEdges, 33
 - GetEdgeIndex, 35
 - mEdges, 36
 - mUnselectDrawingBrush, 36
 - Node, 36
 - RenderEdges, 35
 - Select, 35
 - Selected, 36
 - SetDrawingBrush, 35
 - Unselect, 35
 - UseCaseNode, 33
- UseCaseAnalyser::GraphVisualiser::UseCaseGraph↔
 - Visualiser
 - AddEdge, 24
 - AddNode, 24

- Background_OnPreviewMouseLeftButtonDown, 26
- CanvasScrollViewer_OnMouseWheel, 26
- CanvasSizeUpdateTime, 31
- Clear, 26
- ElementHeight, 31
- ElementWidth, 31
- GetPreviousNodeVariantCount, 26
- GraphElement, 32
- GraphElementProperty, 31
- GraphVisualiser_OnMouseDown, 28
- GraphVisualiser_OnMouseMove, 28
- GraphVisualiser_OnMouseUp, 28
- mLastSizeUpdateTime, 31
- mNodePosDict, 31
- mNodes, 31
- mOffsetElementPosition, 31
- mSelectedElement, 31
- RedrawGraph, 29
- ScaleRateZoom, 31
- Scenario, 32
- ScenarioProperty, 31
- ScenarioPropertyChanged, 29
- SetBrushForScenario, 29
- UseCase, 32
- UseCaseGraphVisualiser, 24
- UseCaseProperty, 31
- UseCasePropertyChanged, 30
- VisualiseEdges, 30
- VisualiseGraph, 30
- VisualiseNodes, 30
- UseCaseEdge
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
 - Elements::UseCaseEdge, 20
- UseCaseGraphVisualiser
 - UseCaseAnalyser::GraphVisualiser::UseCase↔
 - GraphVisualiser, 24
- UseCaseGraphVisualiser.xaml.cs, 38
- UseCaseNode
 - UseCaseAnalyser::GraphVisualiser::Drawing↔
 - Elements::UseCaseNode, 33
- UseCaseProperty
 - UseCaseAnalyser::GraphVisualiser::UseCase↔
 - GraphVisualiser, 31
- UseCasePropertyChanged
 - UseCaseAnalyser::GraphVisualiser::UseCase↔
 - GraphVisualiser, 30
- VisualiseEdges
 - UseCaseAnalyser::GraphVisualiser::UseCase↔
 - GraphVisualiser, 30
- VisualiseGraph
 - UseCaseAnalyser::GraphVisualiser::UseCase↔
 - GraphVisualiser, 30
- VisualiseNodes
 - UseCaseAnalyser::GraphVisualiser::UseCase↔
 - GraphVisualiser, 30