

UseCaseAnalyser.Model

Erzeugt von Doxygen 1.8.9.1

Die Jun 30 2015 08:49:32

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.Model	9
5.3	Paket UseCaseAnalyser.Model.Model	9
5.3.1	Dokumentation der Aufzählungstypen	10
5.3.1.1	NodeAttributes	10
5.3.1.2	UseCaseAttributes	10
5.4	Paket UseCaseAnalyser.Model.ViewModel	11
5.4.1	Dokumentation der Aufzählungstypen	11
5.4.1.1	FileDialogType	11
5.4.1.2	MessageType	11
6	Klassen-Dokumentation	13
6.1	UseCaseAnalyser.Model.ViewModel.AsyncCommand Klassenreferenz	13
6.1.1	Ausführliche Beschreibung	14
6.1.2	Beschreibung der Konstruktoren und Destruktoren	14
6.1.2.1	AsyncCommand	14
6.1.3	Dokumentation der Elementfunktionen	14
6.1.3.1	CanExecute	14
6.1.3.2	Execute	15
6.1.4	Dokumentation der Datenelemente	15

6.1.4.1	mCanExecuteFunc	15
6.1.4.2	mExecuteAction	15
6.1.4.3	mlsExecuting	15
6.1.4.4	mOnError	15
6.1.4.5	UiTaskFactory	15
6.1.5	Dokumentation der Propertys	15
6.1.5.1	CanExecuteChanged	15
6.2	UseCaseAnalyser.Model.ViewModel.DialogViewModel Klassenreferenz	15
6.2.1	Ausführliche Beschreibung	17
6.2.2	Beschreibung der Konstruktoren und Destruktoren	17
6.2.2.1	DialogViewModel	17
6.2.2.2	DialogViewModel	17
6.2.3	Dokumentation der Elementfunktionen	18
6.2.3.1	OnError	18
6.2.3.2	OnPropertyChanged	18
6.2.4	Dokumentation der Datenelemente	18
6.2.4.1	mExportAllScenarioMatrices	18
6.2.4.2	mExportScenarioMatrix	18
6.2.4.3	mOpenAboutView	18
6.2.4.4	mOpenLogfile	18
6.2.4.5	mOpenReportView	18
6.2.4.6	mOpenWordFile	18
6.2.4.7	mRefreshGraph	18
6.2.4.8	mSelectedScenario	18
6.2.4.9	mSelectedUseCaseGraph	18
6.2.4.10	mUseCaseGraphs	18
6.2.4.11	mViewAbstraction	18
6.2.5	Dokumentation der Propertys	18
6.2.5.1	ExportAllScenarioMatrices	18
6.2.5.2	ExportScenarioMatrix	18
6.2.5.3	LatestWordImportReport	19
6.2.5.4	OpenAboutView	19
6.2.5.5	OpenLogfile	19
6.2.5.6	OpenReportView	19
6.2.5.7	OpenWordFile	19
6.2.5.8	RefreshGraph	19
6.2.5.9	SelectedScenario	19
6.2.5.10	SelectedUseCaseGraph	19
6.2.5.11	UseCaseGraphs	19
6.2.6	Ereignisdokumentation	19

6.2.6.1	PropertyChanged	19
6.3	UseCaseAnalyser.Model.Model.HiddenAttribute Klassenreferenz	20
6.3.1	Ausführliche Beschreibung	20
6.3.2	Beschreibung der Konstruktoren und Destruktoren	20
6.3.2.1	HiddenAttribute	20
6.4	UseCaseAnalyser.Model.ViewModel.IDialogView Schnittstellenreferenz	21
6.4.1	Ausführliche Beschreibung	21
6.4.2	Dokumentation der Elementfunktionen	21
6.4.2.1	OpenAboutView	21
6.4.2.2	OpenFileDialog	21
6.4.2.3	OpenMessageBox	22
6.4.2.4	OpenReportResult	23
6.4.2.5	RedrawGraph	23
6.5	UseCaseAnalyser.Model.Model.Report Klassenreferenz	23
6.5.1	Ausführliche Beschreibung	24
6.5.2	Dokumentation der Aufzählungstypen	24
6.5.2.1	Entrytype	24
6.5.3	Dokumentation der Elementfunktionen	24
6.5.3.1	AddReportEntry	24
6.5.3.2	GetEntriesByTag	25
6.5.4	Dokumentation der Datenelemente	25
6.5.4.1	mErrorReportEntries	25
6.5.4.2	mLogReportEntries	25
6.5.4.3	mWarningReportEntries	26
6.5.5	Dokumentation der Propertys	26
6.5.5.1	ErrorReportEntries	26
6.5.5.2	LogReportEntries	26
6.5.5.3	WarningReportEntries	26
6.6	UseCaseAnalyser.Model.Model.Report.ReportEntry Klassenreferenz	26
6.6.1	Ausführliche Beschreibung	26
6.6.2	Beschreibung der Konstruktoren und Destruktoren	27
6.6.2.1	ReportEntry	27
6.6.3	Dokumentation der Propertys	28
6.6.3.1	Content	28
6.6.3.2	Heading	28
6.6.3.3	Tag	28
6.6.3.4	Type	28
6.7	UseCaseAnalyser.Model.Model.ScenarioMatrixCreator Klassenreferenz	28
6.7.1	Ausführliche Beschreibung	29
6.7.2	Dokumentation der Elementfunktionen	29

6.7.2.1	CountVariants	29
6.7.2.2	CreateScenarioMatrix	29
6.7.2.3	CreateScenarios	29
6.7.2.4	ExtendOrderAttribute	29
6.7.2.5	FindStartNode	29
6.7.2.6	GetNodeNumber	29
6.7.2.7	IsAlternativeNode	29
6.7.2.8	IsEndNode	30
6.7.2.9	IsVariantEntry	30
6.7.3	Dokumentation der Datenelemente	30
6.7.3.1	COrder	30
6.7.3.2	CScenarioName	30
6.7.3.3	CUseCase	30
6.8	UseCaseAnalyser.Model.Model.ScenarioMatrixExporter Klassenreferenz	30
6.8.1	Ausführliche Beschreibung	30
6.8.2	Dokumentation der Elementfunktionen	31
6.8.2.1	CreateAndFillExcelPages	31
6.8.2.2	ExportScenarioMatrix	32
6.8.2.3	GetExcelAdressFromXY	32
6.8.2.4	NodeIsNodeInBranch	32
6.8.2.5	ValidateFile	32
6.8.2.6	WriteScenarios	32
6.8.3	Dokumentation der Datenelemente	32
6.8.3.1	COrder	32
6.8.3.2	ExcelExtension	32
6.9	UseCaseAttributeExtensions Klassenreferenz	33
6.9.1	Ausführliche Beschreibung	33
6.9.2	Dokumentation der Elementfunktionen	33
6.9.2.1	Attribute	33
6.9.2.2	Attribute	34
6.9.2.3	AttributeName	34
6.9.2.4	AttributeName	34
6.9.2.5	AttributeValue< T >	35
6.9.2.6	AttributeValue< T >	35
6.9.2.7	ByName	35
6.9.2.8	CreateAttribute< T >	36
6.9.2.9	CreateAttribute< T >	37
6.9.2.10	CreateAttribute< T Value >	37
6.10	UseCaseAnalyser.Model.Model.UseCaseGraph Klassenreferenz	37
6.10.1	Ausführliche Beschreibung	39

6.10.2	Dokumentation der Aufzählungstypen	39
6.10.2.1	NodeTypeAttribute	39
6.10.3	Beschreibung der Konstruktoren und Destruktoren	39
6.10.3.1	UseCaseGraph	39
6.10.4	Dokumentation der Elementfunktionen	40
6.10.4.1	InitAttribute< T >	40
6.10.4.2	RecalculateScenarios	40
6.10.4.3	ToString	40
6.10.5	Dokumentation der Datenelemente	40
6.10.5.1	mScenarios	40
6.10.5.2	NodeAttributeNames	40
6.10.5.3	UseCaseGraphAttributeNames	40
6.10.6	Dokumentation der Propertys	41
6.10.6.1	Scenarios	41
6.11	UseCaseAnalyser.Model.Model.WordImporter Klassenreferenz	41
6.11.1	Ausführliche Beschreibung	42
6.11.2	Dokumentation der Elementfunktionen	43
6.11.2.1	FixedOpen	43
6.11.2.2	FixInvalidUri	43
6.11.2.3	FixUri	43
6.11.2.4	ImportUseCases	43
6.11.2.5	ImportUseCases	43
6.11.2.6	IsUseCaseTableFormat	44
6.11.2.7	TryGetHorizontalContent	45
6.11.2.8	TryGetNormalRoutineAndSeqVars	45
6.11.2.9	TryGetUseCaseId	46
6.11.2.10	TryGetUseCaseName	46
6.11.2.11	TryGetVerticalContent	47
6.11.2.12	TryReadInUseCase	47
6.11.3	Dokumentation der Datenelemente	48
6.11.3.1	actUseCaseId	48
6.11.3.2	actUseCases	48
6.11.3.3	SequenceJump	48
6.11.3.4	UseCaseEnd	48
6.11.3.5	UseCaseJump	48
6.11.3.6	wordImporterReport	48
7	Datei-Dokumentation	49
7.1	-global-/UseCaseAttributeExtensions.cs-Dateireferenz	49
7.1.1	Dokumentation der benutzerdefinierten Typen	49

7.1.1.1	Attribute	49
7.2	Model/HiddenAttribute.cs-Dateireferenz	49
7.3	Model/Report.cs-Dateireferenz	49
7.4	Model/ScenarioMatrixCreator.cs-Dateireferenz	50
7.4.1	Dokumentation der benutzerdefinierten Typen	50
7.4.1.1	Attribute	50
7.5	Model/ScenarioMatrixExporter.cs-Dateireferenz	50
7.6	Model/UseCaseGraph.cs-Dateireferenz	50
7.7	Model/WordImporter.cs-Dateireferenz	51
7.7.1	Dokumentation der benutzerdefinierten Typen	51
7.7.1.1	Attribute	51
7.7.1.2	Table	51
7.8	Properties/AssemblyInfo.cs-Dateireferenz	51
7.9	ViewModel/AsyncCommand.cs-Dateireferenz	51
7.10	ViewModel/DialogViewModel.cs-Dateireferenz	52
7.11	ViewModel/IDialogView.cs-Dateireferenz	52
Index		53

Kapitel 1

Verzeichnis der Namensbereiche

1.1 Pakete

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

UseCaseAnalyser	9
UseCaseAnalyser.Model	9
UseCaseAnalyser.Model.Model	9
UseCaseAnalyser.Model.ViewModel	11

Kapitel 2

Hierarchie-Verzeichnis

2.1 Klassenhierarchie

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

Attribute	
UseCaseAnalyser.Model.Model.HiddenAttribute	20
Graph	
UseCaseAnalyser.Model.Model.UseCaseGraph	37
ICommand	
UseCaseAnalyser.Model.ViewModel.AsyncCommand	13
UseCaseAnalyser.Model.ViewModel.IDialogView	21
INotifyPropertyChanged	
UseCaseAnalyser.Model.ViewModel.DialogViewModel	15
UseCaseAnalyser.Model.Model.Report	23
UseCaseAnalyser.Model.Model.Report.ReportEntry	26
UseCaseAnalyser.Model.Model.ScenarioMatrixCreator	28
UseCaseAnalyser.Model.Model.ScenarioMatrixExporter	30
UseCaseAttributeExtensions	33
UseCaseAnalyser.Model.Model.WordImporter	41

Kapitel 3

Klassen-Verzeichnis

3.1 Auflistung der Klassen

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

UseCaseAnalyser.Model.ViewModel.AsyncCommand	
implementation of the icommand interface. used to bind to from view side	13
UseCaseAnalyser.Model.ViewModel.DialogViewModel	
main view model of the application provides all properties which will be displayed in view . . .	15
UseCaseAnalyser.Model.Model.HiddenAttribute	
a marker class for attributes to filter some attributes from displaying in the view	20
UseCaseAnalyser.Model.ViewModel.IDialogView	
abstraction of the dialog view used to execute view actions from viewmodel side	21
UseCaseAnalyser.Model.Model.Report	
The report class	23
UseCaseAnalyser.Model.Model.Report.ReportEntry	
Data holder	26
UseCaseAnalyser.Model.Model.ScenarioMatrixCreator	
class to create the scenarios for a use case graph	28
UseCaseAnalyser.Model.Model.ScenarioMatrixExporter	
class to export the scenario matrix to a file	30
UseCaseAttributeExtensions	
extension methods for easier attribute access	33
UseCaseAnalyser.Model.Model.UseCaseGraph	
class to represent a use case.	37
UseCaseAnalyser.Model.Model.WordImporter	
Imports use case graphs from a word document	41

Kapitel 4

Datei-Verzeichnis

4.1 Auflistung der Dateien

Hier folgt die Aufzählung aller Dateien mit einer Kurzbeschreibung:

-global-/UseCaseAttributeExtensions.cs	49
Model/HiddenAttribute.cs	49
Model/Report.cs	49
Model/ScenarioMatrixCreator.cs	50
Model/ScenarioMatrixExporter.cs	50
Model/UseCaseGraph.cs	50
Model/WordImporter.cs	51
Properties/AssemblyInfo.cs	51
ViewModel/AsyncCommand.cs	51
ViewModel/DialogViewModel.cs	52
ViewModel/IDialogView.cs	52

Kapitel 5

Dokumentation der Namensbereiche

5.1 Paket UseCaseAnalyser

Namensbereiche

- package [Model](#)

5.2 Paket UseCaseAnalyser.Model

Namensbereiche

- package [Model](#)
- package [ViewModel](#)

5.3 Paket UseCaseAnalyser.Model.Model

Klassen

- class [HiddenAttribute](#)
a marker class for attributes to filter some attributes from displaying in the view
- class [Report](#)
The report class
- class [ScenarioMatrixCreator](#)
class to create the scenarios for a use case graph
- class [ScenarioMatrixExporter](#)
class to export the scenario matrix to a file
- class [UseCaseGraph](#)
class to represent a use case.
- class [WordImporter](#)
Imports use case graphs from a word document

Aufzählungen

- enum [UseCaseAttributes](#) {
 [UseCaseAttributes.Name](#) = 0, [UseCaseAttributes.Id](#), [UseCaseAttributes.Priority](#), [UseCaseAttributes.Description](#),
 [UseCaseAttributes.PreCondition](#), [UseCaseAttributes.PostCondition](#), [UseCaseAttributes.NormalRoutine](#),

UseCaseAttributes.SequenceVariation,
 UseCaseAttributes.SpecialRequirements, UseCaseAttributes.OpenPoints, UseCaseAttributes.Traverse↔
 VariantCount, UseCaseAttributes.TraverseLoopCount }

The access enum to the array UseCaseGraphAttributeNames

- enum **NodeAttributes** {
 NodeAttributes.NormalIndex, NodeAttributes.VariantIndex, NodeAttributes.VarSeqStep, NodeAttributes.↔
 Description,
 NodeAttributes.NodeType }

This enum is used to access the attribute names of the string array NodeAttributeNames

5.3.1 Dokumentation der Aufzählungstypen

5.3.1.1 enum UseCaseAnalyser.Model.Model.NodeAttributes

This enum is used to access the attribute names of the string array NodeAttributeNames

Aufzählungswerte

NormalIndex the index of the normal routine, e.g. "2"

VariantIndex the variant identifier, e.g. "a"

VarSeqStep the variant sequence step, e.g. "1."

Description The description of the node

NodeType The type of the node which are start, end, jump nodes, etc.

5.3.1.2 enum UseCaseAnalyser.Model.Model.UseCaseAttributes

The access enum to the array UseCaseGraphAttributeNames

Aufzählungswerte

Name the name of the use case, e.g. "UseCase-Dokument Importieren"

Id the id of the use case, e.g. "UC-1"

Priority the priority of the use case, e.g. "hoch"

Description the description of the use case, e.g. "Der Anwender möchte ein vorliegendes Word Dokument, welches UseCases beinhaltet in das Tool importieren."

PreCondition the pre condition of the use case, e.g. "Das Dokument (.docx) hat das richtige Format und ist nicht beschaedigt."

PostCondition the post condition of the use case, e.g. "Die UseCases existieren als Datenstruktur und können weiterverarbeitet werden."

NormalRoutine the normal routine of the use case

SequenceVariation the sequence variation of the use case

SpecialRequirements the special requirements of the use case, e.g. "keine"

OpenPoints the open points of the use case, e.g. "Soll der Anwender mehrere Dateien auswählen können, die eingelesen werden sollen?"

TraverseVariantCount how many variants should be traversed in one scenario

TraverseLoopCount how often loops should be traversed in the scenarios

5.4 Paket UseCaseAnalyser.Model.ViewModel

Klassen

- class [AsyncCommand](#)
implementation of the ICommand interface. used to bind to from view side
- class [DialogViewModel](#)
main view model of the application provides all properties which will be displayed in view
- interface [IDialogView](#)
abstraction of the dialog view used to execute view actions from viewmodel side

Aufzählungen

- enum [MessageType](#) { [MessageType.Information](#), [MessageType.Warning](#), [MessageType.Error](#) }
enum for the different message types to be displayed in message boxes
- enum [FileDialogType](#) { [FileDialogType.Open](#), [FileDialogType.Save](#) }
enum for the different file dialog types

5.4.1 Dokumentation der Aufzählungstypen

5.4.1.1 enum UseCaseAnalyser.Model.ViewModel.FileDialogType

enum for the different file dialog types

Aufzählungswerte

Open dialog type to open a document

Save dialog type to save a document

5.4.1.2 enum UseCaseAnalyser.Model.ViewModel.MessageType

enum for the different message types to be displayed in message boxes

Aufzählungswerte

Information message type to display an information

Warning message type to display a warning

Error message type to display an error

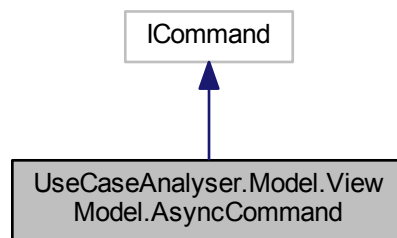
Kapitel 6

Klassen-Dokumentation

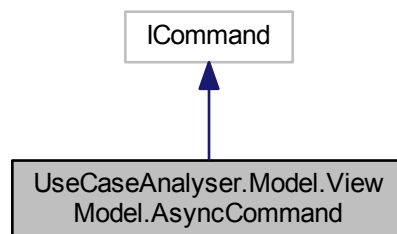
6.1 UseCaseAnalyser.Model.ViewModel.AsyncCommand Klassenreferenz

implementation of the icommand interface. used to bind to from view side

Klassendiagramm für UseCaseAnalyser.Model.ViewModel.AsyncCommand:



Zusammengehörigkeiten von UseCaseAnalyser.Model.ViewModel.AsyncCommand:



Öffentliche Methoden

- [AsyncCommand](#) (Action< object > executeAction, Func< object, bool > canExecuteFunc, Action< Exception > onError)
creates a new command to bind to from the gui
- bool [CanExecute](#) (object parameter)
checks if the command is currently executable
- void [Execute](#) (object parameter)
executes the action of the command

Property

- EventHandler [CanExecuteChanged](#)
invoked if the commandmanager detects action which might change the executable state -> can execute will be invoked

Private Attribute

- readonly Action< object > [mExecuteAction](#)
- readonly Func< object, bool > [mCanExecuteFunc](#)
- readonly Action< Exception > [mOnError](#)
- bool [mIsExecuting](#)

Statische, private Attribute

- static readonly TaskFactory [UiTaskFactory](#) = new TaskFactory(TaskScheduler.FromCurrentSynchronizationContext())

6.1.1 Ausführliche Beschreibung

implementation of the ICommand interface. used to bind to from view side

6.1.2 Beschreibung der Konstruktoren und Destruktoren

6.1.2.1 UseCaseAnalyser.Model.ViewModel.AsyncCommand.AsyncCommand (Action< object > executeAction, Func< object, bool > canExecuteFunc, Action< Exception > onError)

creates a new command to bind to from the gui

Parameter

<i>executeAction</i>	action to execute on command execute
<i>canExecuteFunc</i>	function to determine weather the action can be executed
<i>onError</i>	action to run if the execute action throws an exception

6.1.3 Dokumentation der Elementfunktionen

6.1.3.1 bool UseCaseAnalyser.Model.ViewModel.AsyncCommand.CanExecute (object parameter)

checks if the command is currently executable

Parameter

<i>parameter</i>	parameter which can be passed from the view
------------------	---

Rückgabe

weather the command is executable

6.1.3.2 void UseCaseAnalyser.Model.ViewModel.AsyncCommand.Execute (object *parameter*)

executes the action of the command

Parameter

<i>parameter</i>	action parameter which can be passed from the view
------------------	--

6.1.4 Dokumentation der Datenelemente

6.1.4.1 readonly Func<object, bool> UseCaseAnalyser.Model.ViewModel.AsyncCommand.mCanExecuteFunc
[private]

6.1.4.2 readonly Action<object> UseCaseAnalyser.Model.ViewModel.AsyncCommand.mExecuteAction [private]

6.1.4.3 bool UseCaseAnalyser.Model.ViewModel.AsyncCommand.mIsExecuting [private]

6.1.4.4 readonly Action<Exception> UseCaseAnalyser.Model.ViewModel.AsyncCommand.mOnError [private]

6.1.4.5 readonly TaskFactory UseCaseAnalyser.Model.ViewModel.AsyncCommand.UiTaskFactory = new
TaskFactory(TaskScheduler.FromCurrentSynchronizationContext()) [static], [private]

6.1.5 Dokumentation der Propertys

6.1.5.1 EventHandler UseCaseAnalyser.Model.ViewModel.AsyncCommand.CanExecuteChanged [add], [remove]

invoked if the commandmanager detects action which might change the executable state -> can execute will be invoked

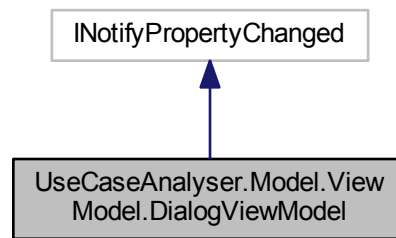
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- ViewModel/[AsyncCommand.cs](#)

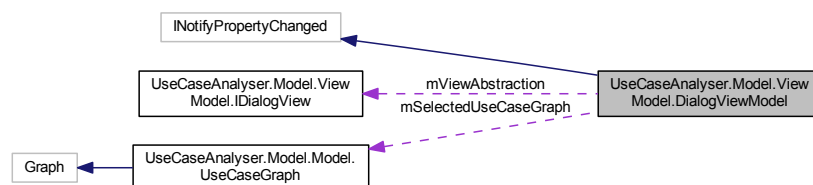
6.2 UseCaseAnalyser.Model.ViewModel.DialogViewModel Klassenreferenz

main view model of the application provides all properties which will be displayed in view

Klassendiagramm für UseCaseAnalyser.Model.ViewModel.DialogViewModel:



Zusammengehörigkeiten von UseCaseAnalyser.Model.ViewModel.DialogViewModel:



Öffentliche Methoden

- [DialogViewModel \(\)](#)
creates a new dialogviewmodel without interface of the view (for tests and wpf designer)
- [DialogViewModel \(IDialogView viewAbstraction\)](#)
creates a new dialogviewmodel. view specific actions can be invoked over the view interface

Geschützte Methoden

- virtual void [OnPropertyChanged](#) ([CallerMemberName] string propertyName=null)
fires the property changed for the given property name

Propertys

- IEnumerable< [UseCaseGraph](#) > [UseCaseGraphs](#) [get, private set]
all use cases which are currently saved (have been read by the word importer)
- [UseCaseGraph SelectedUseCaseGraph](#) [get, set]
the currently selected graph from the view -> set via binding
- IGraph [SelectedScenario](#) [get, set]
the currently selection scenario from the view -> set via binding
- [Report LatestWordImportReport](#) [get, private set]
the latest word import report gotten from the word importer

- ICommand [OpenWordFile](#) [get]
opens a word file and tries to read in the use cases
- ICommand [ExportScenarioMatrix](#) [get]
exports the scenarios from the currently selected use case
- ICommand [ExportAllScenarioMatrices](#) [get]
exports the scenarios from all use cases
- ICommand [OpenLogfile](#) [get]
opens the logfile as seperate process
- ICommand [OpenReportView](#) [get]
opens the latest word import report view in its seperate window
- ICommand [OpenAboutView](#) [get]
opens the about view in its seperate window
- ICommand [RefreshGraph](#) [get]
refreshes the current use case graph visualization

Ereignisse

- PropertyChangedEventHandler [PropertyChanged](#)
invoked to notify the gui about changed of properties

Private Methoden

- void [OnError](#) (Exception ex, string customText=null)

Private Attribute

- readonly [IDialogView](#) [mViewAbstraction](#)
- IEnumerable< [UseCaseGraph](#) > [mUseCaseGraphs](#)
- [UseCaseGraph](#) [mSelectedUseCaseGraph](#)
- IGraph [mSelectedScenario](#)
- ICommand [mExportScenarioMatrix](#)
- ICommand [mExportAllScenarioMatrices](#)
- ICommand [mOpenWordFile](#)
- ICommand [mOpenLogfile](#)
- ICommand [mOpenReportView](#)
- ICommand [mOpenAboutView](#)
- ICommand [mRefreshGraph](#)

6.2.1 Ausführliche Beschreibung

main view model of the application provides all properties which will be displayed in view

6.2.2 Beschreibung der Konstruktoren und Destruktoren

6.2.2.1 UseCaseAnalyser.Model.ViewModel.DialogViewModel.DialogViewModel ()

creates a new dialogviewmodel without interface of the view (for tests and wpf designer)

6.2.2.2 UseCaseAnalyser.Model.ViewModel.DialogViewModel.DialogViewModel ([IDialogView](#) *viewAbstraction*)

creates a new dialogviewmodel. view specific actions can be invoked over the view interface

Parameter

<i>viewAbstraction</i>	interface abstraction of the view
------------------------	-----------------------------------

6.2.3 Dokumentation der Elementfunktionen

6.2.3.1 `void UseCaseAnalyser.Model.ViewModel.DialogViewModel.OnError (Exception ex, string customText = null)`
[private]

6.2.3.2 `virtual void UseCaseAnalyser.Model.ViewModel.DialogViewModel.OnPropertyChanged ([CallerMemberName] string propertyName = null)` [protected],[virtual]

fires the property changed for the given property name

Parameter

<i>propertyName</i>	
---------------------	--

6.2.4 Dokumentation der Datenelemente

6.2.4.1 `ICommand UseCaseAnalyser.Model.ViewModel.DialogViewModel.mExportAllScenarioMatrices` [private]

6.2.4.2 `ICommand UseCaseAnalyser.Model.ViewModel.DialogViewModel.mExportScenarioMatrix` [private]

6.2.4.3 `ICommand UseCaseAnalyser.Model.ViewModel.DialogViewModel.mOpenAboutView` [private]

6.2.4.4 `ICommand UseCaseAnalyser.Model.ViewModel.DialogViewModel.mOpenLogfile` [private]

6.2.4.5 `ICommand UseCaseAnalyser.Model.ViewModel.DialogViewModel.mOpenReportView` [private]

6.2.4.6 `ICommand UseCaseAnalyser.Model.ViewModel.DialogViewModel.mOpenWordFile` [private]

6.2.4.7 `ICommand UseCaseAnalyser.Model.ViewModel.DialogViewModel.mRefreshGraph` [private]

6.2.4.8 `IGraph UseCaseAnalyser.Model.ViewModel.DialogViewModel.mSelectedScenario` [private]

6.2.4.9 `UseCaseGraph UseCaseAnalyser.Model.ViewModel.DialogViewModel.mSelectedUseCaseGraph` [private]

6.2.4.10 `IEnumerable<UseCaseGraph> UseCaseAnalyser.Model.ViewModel.DialogViewModel.mUseCaseGraphs`
[private]

6.2.4.11 `readonly IDialogView UseCaseAnalyser.Model.ViewModel.DialogViewModel.mViewAbstraction` [private]

6.2.5 Dokumentation der Propertys

6.2.5.1 `ICommand UseCaseAnalyser.Model.ViewModel.DialogViewModel.ExportAllScenarioMatrices` [get]

exports the scenarios from all use cases

enabled if: any use case is imported

6.2.5.2 `ICommand UseCaseAnalyser.Model.ViewModel.DialogViewModel.ExportScenarioMatrix` [get]

exports the scenarios from the currently selected use case

enabled if: a use case is selected

6.2.5.3 Report UseCaseAnalyser.Model.ViewModel.DialogViewModel.LatestWordImportReport [get], [private set]

the latest word import report gotten from the word importer

6.2.5.4 ICommand UseCaseAnalyser.Model.ViewModel.DialogViewModel.OpenAboutView [get]

opens the about view in its seperate window

6.2.5.5 ICommand UseCaseAnalyser.Model.ViewModel.DialogViewModel.OpenLogfile [get]

opens the logfile as seperate process

enabled if: the logfile exists

6.2.5.6 ICommand UseCaseAnalyser.Model.ViewModel.DialogViewModel.OpenReportView [get]

opens the latest word import report view in its seperate window

enabled if: there is a latest word import report

6.2.5.7 ICommand UseCaseAnalyser.Model.ViewModel.DialogViewModel.OpenWordFile [get]

opens a word file and tries to read in the use cases

6.2.5.8 ICommand UseCaseAnalyser.Model.ViewModel.DialogViewModel.RefreshGraph [get]

refreshes the current use case graph visualization

6.2.5.9 IGraph UseCaseAnalyser.Model.ViewModel.DialogViewModel.SelectedScenario [get], [set]

the currently selection scenario from the view → set via binding

6.2.5.10 UseCaseGraph UseCaseAnalyser.Model.ViewModel.DialogViewModel.SelectedUseCaseGraph [get], [set]

the currently selected graph from the view → set via binding

6.2.5.11 IEnumerable<UseCaseGraph> UseCaseAnalyser.Model.ViewModel.DialogViewModel.UseCaseGraphs [get], [private set]

all use cases which are currently saved (have been read by the word importer)

6.2.6 Ereignisdokumentation

6.2.6.1 PropertyChangedEventHandler UseCaseAnalyser.Model.ViewModel.DialogViewModel.PropertyChanged

invoked to notify the gui about changed of properties

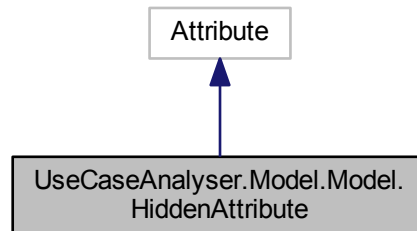
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- ViewModel/[DialogViewModel.cs](#)

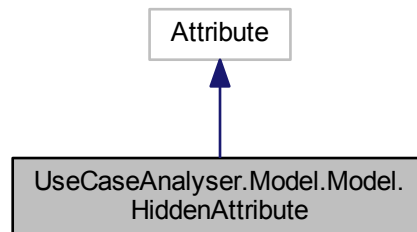
6.3 UseCaseAnalyser.Model.Model.HiddenAttribute Klassenreferenz

a marker class for attributes to filter some attributes from displaying in the view

Klassendiagramm für UseCaseAnalyser.Model.Model.HiddenAttribute:



Zusammengehörigkeiten von UseCaseAnalyser.Model.Model.HiddenAttribute:



Öffentliche Methoden

- [HiddenAttribute](#) (string name, object value)
initializes the attribute with its name and its value

6.3.1 Ausführliche Beschreibung

a marker class for attributes to filter some attributes from displaying in the view

6.3.2 Beschreibung der Konstruktoren und Destruktoren

6.3.2.1 UseCaseAnalyser.Model.Model.HiddenAttribute.HiddenAttribute (string name, object value)

initializes the attribute with its name and its value

Parameter

<i>name</i>	the name of the attribute
<i>value</i>	the value of the attribute

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Model/[HiddenAttribute.cs](#)

6.4 UseCaseAnalyser.Model.ViewModel.IDialogView Schnittstellenreferenz

abstraction of the dialog view used to execute view actions from viewmodel side

Öffentliche Methoden

- FileInfo [OpenFileDialog](#) (string filter, [FileDialogType](#) dialogType, string predefinedName=null)
opens a file dialog and returns the file
- void [OpenMessageBox](#) (string header, string content, [MessageType](#) messageType)
opens a message box with the given parameters
- void [OpenReportResult](#) ([Report](#) viewModel)
opens the report view
- void [OpenAboutView](#) ()
Opens the about box
- void [RedrawGraph](#) ()
lets the graph visualizer redraw the graph

6.4.1 Ausführliche Beschreibung

abstraction of the dialog view used to execute view actions from viewmodel side

6.4.2 Dokumentation der Elementfunktionen

6.4.2.1 void UseCaseAnalyser.Model.ViewModel.IDialogView.OpenAboutView ()

Opens the about box

6.4.2.2 FileInfo UseCaseAnalyser.Model.ViewModel.IDialogView.OpenFileDialog (string filter, FileDialogType dialogType, string predefinedName = null)

opens a file dialog and returns the file

Parameter

<i>filter</i>	filter of the files
<i>dialogType</i>	dialog type (open or save)
<i>predefinedName</i>	the default file name

Rückgabe

the file which has been selected

6.4.2.3 void UseCaseAnalyser.Model.ViewModel.IDialogView.OpenMessageBox (string *header*, string *content*,
MessageType *messageType*)

opens a message box with the given parameters

Parameter

<i>header</i>	header of the message box
<i>content</i>	content of the message box
<i>messageType</i>	type of the message → determines the message box icon

6.4.2.4 void UseCaseAnalyser.Model.ViewModel.IDialogView.OpenReportResult (Report *viewModel*)

opens the report view

Parameter

<i>viewModel</i>	the report which is used as viewmodel of the report view
------------------	--

6.4.2.5 void UseCaseAnalyser.Model.ViewModel.IDialogView.RedrawGraph ()

lets the graph visualizer redraw the graph

Die Dokumentation für diese Schnittstelle wurde erzeugt aufgrund der Datei:

- ViewModel/[IDialogView.cs](#)

6.5 UseCaseAnalyser.Model.Model.Report Klassenreferenz

The report class

Klassen

- class [ReportEntry](#)
Data holder

Öffentliche Typen

- enum [Entrytype](#) { [Entrytype.ERROR](#), [Entrytype.WARNING](#), [Entrytype.LOG](#), [Entrytype.DEFAULT](#) }
The type of the entry

Öffentliche Methoden

- void [AddReportEntry](#) ([ReportEntry](#) entry)
Adds an entry to the report
- List< [ReportEntry](#) > [GetEntriesByTag](#) (string tag, [Entrytype](#) type=[Entrytype.DEFAULT](#))
Returns all report entries with the specified tag

Propertys

- [ReportEntry\[\]](#) [ErrorReportEntries](#) [get]
All error report entries
- [ReportEntry\[\]](#) [WarningReportEntries](#) [get]
All warning report entries
- [ReportEntry\[\]](#) [LogReportEntries](#) [get]
All log report entries

Private Attribute

- readonly List< [ReportEntry](#) > mErrorReportEntries = new List<[ReportEntry](#)>()

All error entries

- readonly List< [ReportEntry](#) > mWarningReportEntries = new List<[ReportEntry](#)>()

All warning entries

- readonly List< [ReportEntry](#) > mLogReportEntries = new List<[ReportEntry](#)>()

All log entries

6.5.1 Ausführliche Beschreibung

The report class

6.5.2 Dokumentation der Aufzählungstypen

6.5.2.1 enum UseCaseAnalyser.Model.Model.Report.Entrytype

The type of the entry

Aufzählungswerte

ERROR error report entry

WARNING warning report entry

LOG information report entry

DEFAULT default report entry

6.5.3 Dokumentation der Elementfunktionen

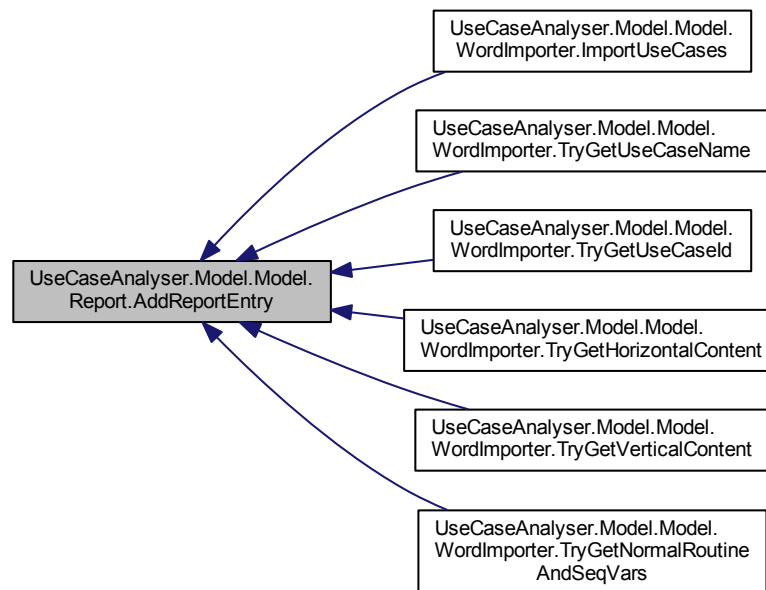
6.5.3.1 void UseCaseAnalyser.Model.Model.Report.AddReportEntry ([ReportEntry](#) entry)

Adds an entry to the report

Parameter

<i>entry</i>	
--------------	--

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



6.5.3.2 `List<ReportEntry> UseCaseAnalyser.Model.Model.Report.GetEntriesByTag (string tag, Entrytype type = Entrytype.DEFAULT)`

Returns all report entries with the specified tag

Parameter

<i>tag</i>	
<i>type</i>	optional

Rückgabe

6.5.4 Dokumentation der Datenelemente

6.5.4.1 `readonly List<ReportEntry> UseCaseAnalyser.Model.Model.Report.mErrorReportEntries = new List<ReportEntry>() [private]`

All error entries

6.5.4.2 `readonly List<ReportEntry> UseCaseAnalyser.Model.Model.Report.mLogReportEntries = new List<ReportEntry>() [private]`

All log entries

```
6.5.4.3 readonly List<ReportEntry> UseCaseAnalyser.Model.Model.Report.mWarningReportEntries = new
        List<ReportEntry>() [private]
```

All warning entries

6.5.5 Dokumentation der Propertys

```
6.5.5.1 ReportEntry [] UseCaseAnalyser.Model.Model.Report.ErrorReportEntries [get]
```

All error report entries

```
6.5.5.2 ReportEntry [] UseCaseAnalyser.Model.Model.Report.LogReportEntries [get]
```

All log report entries

```
6.5.5.3 ReportEntry [] UseCaseAnalyser.Model.Model.Report.WarningReportEntries [get]
```

All warning report entries

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Model/[Report.cs](#)

6.6 UseCaseAnalyser.Model.Model.Report.ReportEntry Klassenreferenz

Data holder

Öffentliche Methoden

- [ReportEntry](#) (string heading, string content, [Entrytype](#) type, string tag="")
creates a new report entry with the given parameters

Propertys

- string [Heading](#) [get, private set]
A brief summary of the report entry
- string [Content](#) [get, private set]
A description of the report entry
- [Entrytype](#) Type [get, private set]
The type, e.g. log, error, warning
- string [Tag](#) [get, private set]
The tag of the report entry. Used to be one word. By default the tag of an entry is an empty string

6.6.1 Ausführliche Beschreibung

Data holder

6.6.2 Beschreibung der Konstruktoren und Destruktoren

6.6.2.1 UseCaseAnalyser.Model.Model.Report.ReportEntry.ReportEntry (string *heading*, string *content*, Entrytype *type*, string *tag* = " ")

creates a new report entry with the given parameters

Parameter

<i>heading</i>	heading of the report entry
<i>content</i>	content of the report entry
<i>type</i>	entry type (error, warning, log)
<i>tag</i>	optional additional tag of the report entry

6.6.3 Dokumentation der Propertys

6.6.3.1 `string UseCaseAnalyser.Model.Model.Report.ReportEntry.Content` [get],[private set]

A description of the report entry

6.6.3.2 `string UseCaseAnalyser.Model.Model.Report.ReportEntry.Heading` [get],[private set]

A brief summary of the report entry

6.6.3.3 `string UseCaseAnalyser.Model.Model.Report.ReportEntry.Tag` [get],[private set]

The tag of the report entry. Used to be one word. By default the tag of an entry is an empty string

6.6.3.4 `Entrytype UseCaseAnalyser.Model.Model.Report.ReportEntry.Type` [get],[private set]

The type, e.g. log, error, warning

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Model/[Report.cs](#)

6.7 UseCaseAnalyser.Model.Model.ScenarioMatrixCreator Klassenreferenz

class to create the scenarios for a use case graph

Öffentliche, statische Methoden

- static `IEnumerable< IGraph > CreateScenarios (UseCaseGraph useCaseGraph)`
Creates all scenarios from a Use-Case graph.

Private, statische Methoden

- static `string GetNodeNumber (INode node)`
- static `string ExtendOrderAttribute (string attributeValue, INode nextNode, UseCaseGraph useCaseGraph, IEdge correspondingEdge=null)`
- static `INode FindStartNode (UseCaseGraph graph)`
- static `bool IsEndNode (INode node, UseCaseGraph useCaseGraph)`
- static `bool IsAlternativeNode (INode node)`
- static `int CountVariants (IGraph graph)`
- static `bool IsVariantEntry (IEdge edge)`
- static `IEnumerable< IGraph > CreateScenarioMatrix (INode currentNode, IGraph existingScenario, UseCaseGraph useCaseGraph, int maxVariantTraversions, IDictionary< IEdge, int > numLoopTraversions, int maxLoopTraversions)`

Private Attribute

- const string `CUseCase` = "Scenario of UseCase"
- const string `COrder` = "Order of Visit"
- const string `CScenarioName` = "Name"

6.7.1 Ausführliche Beschreibung

class to create the scenarios for a use case graph

6.7.2 Dokumentation der Elementfunktionen

6.7.2.1 static int UseCaseAnalyser.Model.Model.ScenarioMatrixCreator.CountVariants (IGraph *graph*) [static], [private]

6.7.2.2 static IEnumerable<IGraph> UseCaseAnalyser.Model.Model.ScenarioMatrixCreator.CreateScenarioMatrix (INode *currentNode*, IGraph *existingScenario*, UseCaseGraph *useCaseGraph*, int *maxVariantTraversions*, IDictionary< IEdge, int > *numLoopTraversions*, int *maxLoopTraversions*) [static], [private]

6.7.2.3 static IEnumerable<IGraph> UseCaseAnalyser.Model.Model.ScenarioMatrixCreator.CreateScenarios (UseCaseGraph *useCaseGraph*) [static]

Creates all scenarios from a Use-Case graph.

Parameter

<i>useCaseGraph</i>	Use-Case graph to get its scenarios from
---------------------	--

Rückgabe

scenario matrix (as array of graphs → scenarios)

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



6.7.2.4 static string UseCaseAnalyser.Model.Model.ScenarioMatrixCreator.ExtendOrderAttribute (string *attributeValue*, INode *nextNode*, UseCaseGraph *useCaseGraph*, IEdge *correspondingEdge* = null) [static], [private]

6.7.2.5 static INode UseCaseAnalyser.Model.Model.ScenarioMatrixCreator.FindStartNode (UseCaseGraph *graph*) [static], [private]

6.7.2.6 static string UseCaseAnalyser.Model.Model.ScenarioMatrixCreator.GetNodeNumber (INode *node*) [static], [private]

6.7.2.7 static bool UseCaseAnalyser.Model.Model.ScenarioMatrixCreator.IsAlternativeNode (INode *node*) [static], [private]

6.7.2.8 static bool UseCaseAnalyser.Model.Model.ScenarioMatrixCreator.IsEndNode (INode *node*, UseCaseGraph *useCaseGraph*) [static],[private]

6.7.2.9 static bool UseCaseAnalyser.Model.Model.ScenarioMatrixCreator.IsVariantEntry (IEdge *edge*) [static],[private]

6.7.3 Dokumentation der Datenelemente

6.7.3.1 const string UseCaseAnalyser.Model.Model.ScenarioMatrixCreator.COrder = "Order of Visit" [private]

6.7.3.2 const string UseCaseAnalyser.Model.Model.ScenarioMatrixCreator.CScenarioName = "Name" [private]

6.7.3.3 const string UseCaseAnalyser.Model.Model.ScenarioMatrixCreator.CUseCase = "Scenario of UseCase" [private]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Model/[ScenarioMatrixCreator.cs](#)

6.8 UseCaseAnalyser.Model.Model.ScenarioMatrixExporter Klassenreferenz

class to export the scenario matrix to a file

Öffentliche, statische Methoden

- static void [ExportScenarioMatrix](#) (IEnumerable< [UseCaseGraph](#) > useCaseGraphs, FileInfo file)
exports the scenario matrix of one ore more use case graphs to the specified file (.xlsx)

Private, statische Methoden

- static void [ValidateFile](#) (FileInfo file)
Helper method to check if file complies with preconditions.
- static void [CreateAndFillExcelPages](#) (IEnumerable< [UseCaseGraph](#) > useCaseGraphs, Spreadsheet↔ Document document)
Creates the excel pages.
- static void [WriteScenarios](#) (IEnumerable< IGraph > scenarios, SheetData sd)
Fills the excel page.
- static string [GetExcelAdressFromXY](#) (int x, int y)
- static bool [NodeIsNodeInBranch](#) (string name)

Private Attribute

- const string [ExcelExtension](#) = ".xlsx"
- const string [COrder](#) = "Order of Visit"

6.8.1 Ausführliche Beschreibung

class to export the scenario matrix to a file

6.8.2 Dokumentation der Elementfunktionen

6.8.2.1 static void UseCaseAnalyser.Model.Model.ScenarioMatrixExporter.CreateAndFillExcelPages (IEnumerable< UseCaseGraph > *useCaseGraphs*, SpreadsheetDocument *document*) [static],[private]

Creates the excel pages.

Parameter

<i>useCaseGraphs</i>	The use case graphs.
<i>document</i>	The excel document.

Ausnahmebehandlung

<i>System.InvalidOperationException</i>	Use case is corrupt!
---	----------------------

6.8.2.2 `static void UseCaseAnalyser.Model.Model.ScenarioMatrixExporter.ExportScenarioMatrix (IEnumerable< UseCaseGraph > useCaseGraphs, FileInfo file) [static]`

exports the scenario matrix of one ore more use case graphs to the specified file (.xlsx)

Parameter

<i>useCaseGraphs</i>	use case graph(s) whose scenario matrix should be exported
<i>file</i>	file info of the file to save the scenario matrix to

6.8.2.3 `static string UseCaseAnalyser.Model.Model.ScenarioMatrixExporter.GetExcelAdressFromXY (int x, int y) [static], [private]`

6.8.2.4 `static bool UseCaseAnalyser.Model.Model.ScenarioMatrixExporter.NodesNodeInBranch (string name) [static], [private]`

6.8.2.5 `static void UseCaseAnalyser.Model.Model.ScenarioMatrixExporter.ValidateFile (FileInfo file) [static], [private]`

Helper method to check if file complies with preconditions.

Parameter

<i>file</i>	FileInfo that should be checked.
-------------	----------------------------------

6.8.2.6 `static void UseCaseAnalyser.Model.Model.ScenarioMatrixExporter.WriteScenarios (IEnumerable< IGraph > scenarios, SheetData sd) [static], [private]`

Fills the excel page.

Parameter

<i>scenarios</i>	The scenarios.
<i>sd</i>	The sd.

6.8.3 Dokumentation der Datenelemente

6.8.3.1 `const string UseCaseAnalyser.Model.Model.ScenarioMatrixExporter.COrder = "Order of Visit" [private]`

6.8.3.2 `const string UseCaseAnalyser.Model.Model.ScenarioMatrixExporter.ExcelExtension = ".xlsx" [private]`

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Model/[ScenarioMatrixExporter.cs](#)

6.9 UseCaseAttributeExtensions Klassenreferenz

extension methods for easier attribute access

Öffentliche, statische Methoden

- static T **AttributeValue**< T > (this **UseCaseGraph** source, **UseCaseAttributes** useCaseAttribute)
gets the attribute value represented by the given usecasegraph attribute type throws 'KeyNotFoundException' if the attribute is not contained in the source item's attributes
- static T **AttributeValue**< T > (this **INode** source, **NodeAttributes** nodeAttribute)
gets the attribute value represented by the given nodeAttribute type throws 'KeyNotFoundException' if the attribute is not contained in the source item's attributes
- static **IAttribute Attribute** (this **UseCaseGraph** source, **UseCaseAttributes** usecaseAttribute, bool throw↵Exception=true)
gets the attribute represented by the given usecasegraph attribute type throws 'KeyNotFoundException' if throw↵Exception is true and the attribute is not contained in the source item's attributes
- static **IAttribute Attribute** (this **INode** source, **NodeAttributes** nodeAttribute, bool throwException=true)
gets the attribute represented by the given nodeAttribute type throws 'KeyNotFoundException' if throwException is true and the attribute is not contained in the source item's attributes
- static **IAttribute ByName** (this **IEnumerable**< **IAttribute** > sourceEnumerable, string name, bool throw↵Exception=true)
gets the attribute with the given name from the source enumerable throws 'KeyNotFoundException' if throwException is true and the attribute is not contained in the source item's attributes
- static **IAttribute CreateAttribute**< T > (this **UseCaseAttributes** sourceUsecasegraphAttribute, T attribute↵Value, bool hidden=false)
creates a attribute with the name from the given usecasegraph attribute enum value and the given attributeValue
- static **IAttribute CreateAttribute**< TValue > (this **NodeAttributes** sourceNodeAttribute, TValue attributeValue, bool hidden=false)
creates a attribute with the name from the given nodeAttributes enum value and the given attributeValue
- static string **AttributeName** (this **UseCaseAttributes** sourceUseCaseAttribute)
gets the actual attribute name of the value represented by the enum
- static string **AttributeName** (this **NodeAttributes** sourceNodeAttribute)
gets the actual attribute name of the value represented by the enum

Private, statische Methoden

- static **IAttribute CreateAttribute**< T > (int enumValue, **IList**< string > attributeNameList, T attributeValue, bool hidden)

6.9.1 Ausführliche Beschreibung

extension methods for easier attribute access

6.9.2 Dokumentation der Elementfunktionen

- 6.9.2.1 static **IAttribute UseCaseAttributeExtensions.Attribute** (this **UseCaseGraph** source, **UseCaseAttributes** usecaseAttribute, bool throwException = true) [static]

gets the attribute represented by the given usecasegraph attribute type throws 'KeyNotFoundException' if throw↵Exception is true and the attribute is not contained in the source item's attributes

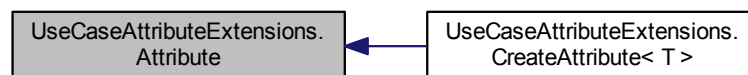
Parameter

<i>source</i>	the node which contains the attributes
<i>usecaseAttribute</i>	the attribute type which should be read
<i>throwException</i>	weather to throw an exception or to return null, if the attribute is not contained in the attributes collection of the usecasegraph

Rückgabe

the attribute value of type T

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



6.9.2.2 static IAttribute UseCaseAttributeExtensions.Attribute (this INode *source*, NodeAttributes *nodeAttribute*, bool *throwException = true*) [static]

gets the attribute represented by the given nodeAttribute type throws 'KeyNotFoundException' if throwException is true and the attribute is not contained in the source item's attributes

Parameter

<i>source</i>	the node which contains the attributes
<i>nodeAttribute</i>	the attribute type which should be read
<i>throwException</i>	weather to throw an exception or to return null, if the attribute is not contained in the attributes collection of the node

Rückgabe

the attribute value of type T

6.9.2.3 static string UseCaseAttributeExtensions.AttributeName (this UseCaseAttributes *sourceUseCaseAttribute*) [static]

gets the actual attribute name of the value represented by the enum

Parameter

<i>sourceUseCaseAttribute</i>	the enum value
-------------------------------	----------------

Rückgabe

the string of the attribute name

6.9.2.4 static string UseCaseAttributeExtensions.AttributeName (this NodeAttributes *sourceNodeAttribute*) [static]

gets the actual attribute name of the value represented by the enum

Parameter

<i>sourceNode↔ Attribute</i>	the enum value
----------------------------------	----------------

Rückgabe

the string of the attribute name

6.9.2.5 `static T UseCaseAttributeExtensions.AttributeValue< T > (this UseCaseGraph source, UseCaseAttributes useCaseAttribute) [static]`

gets the attribute value represented by the given usecasegraph attribute type throws 'KeyNotFoundException' if the attribute is not contained in the source item's attributes

Parameter

<i>source</i>	the node which contains the attributes
<i>useCase↔ Attribute</i>	the attribute type which should be read

Rückgabe

the attribute value of type T

6.9.2.6 `static T UseCaseAttributeExtensions.AttributeValue< T > (this INode source, NodeAttributes nodeAttribute) [static]`

gets the attribute value represented by the given nodeAttribute type throws 'KeyNotFoundException' if the attribute is not contained in the source item's attributes

Parameter

<i>source</i>	the node which contains the attributes
<i>nodeAttribute</i>	the attribute type which should be read

Rückgabe

the attribute value of type T

6.9.2.7 `static IAttribute UseCaseAttributeExtensions.ByName (this IEnumerable< IAttribute > sourceEnumerable, string name, bool throwException = true) [static]`

gets the attribute with the given name from the source enumerable throws 'KeyNotFoundException' if throwException is true and the attribute is not contained in the source item's attributes

Parameter

<i>source↔ Enumerable</i>	the enumerable which should contain the attribute with the given name
<i>name</i>	name of the attribute to search
<i>throwException</i>	weather to throw an exception or to return null, if there is no attribute with the given name.

Rückgabe

the attribute with the given name

6.9.2.8 `static IAttribute UseCaseAttributeExtensions.CreateAttribute< T > (this UseCaseAttributes
sourceUsecasegraphAttribute, T attributeValue, bool hidden = false) [static]`

creates a attribute with the name from the given usecasegraph attribute enum value and the given attributeValue

Parameter

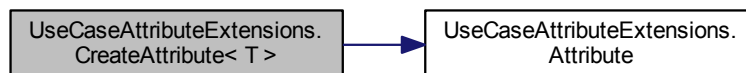
<i>source</i> ↔ <i>Usecasegraph</i> ↔ <i>Attribute</i>	the nodeAttribute enum value
<i>attributeValue</i>	the value of the attribute
<i>hidden</i>	value to determine weather the attribute should be hidden in the gui

Rückgabe

a new attribute

6.9.2.9 `static IAttribute UseCaseAttributeExtensions.CreateAttribute< T > (int enumValue, IList< string > attributeNameList, T attributeValue, bool hidden) [static], [private]`

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



6.9.2.10 `static IAttribute UseCaseAttributeExtensions.CreateAttribute< TValue > (this NodeAttributes sourceNodeAttribute, TValue attributeValue, bool hidden = false) [static]`

creates a attribute with the name from the given nodeAttributes enum value and the given attributeValue

Parameter

<i>sourceNode</i> ↔ <i>Attribute</i>	the nodeAttribute enum value
<i>attributeValue</i>	the value of the attribute
<i>hidden</i>	value to determine weather the attribute should be hidden in the gui

Rückgabe

a new attribute

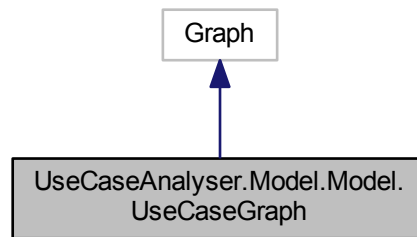
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [-global-/UseCaseAttributeExtensions.cs](#)

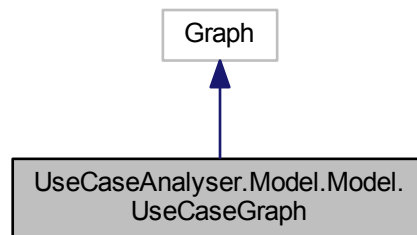
6.10 UseCaseAnalyser.Model.Model.UseCaseGraph Klassenreferenz

class to represent a use case.

Klassendiagramm für UseCaseAnalyser.Model.Model.UseCaseGraph:



Zusammengehörigkeiten von UseCaseAnalyser.Model.Model.UseCaseGraph:



Öffentliche Typen

- enum [NodeTypeAttribute](#) {
[NodeTypeAttribute.StartNode](#), [NodeTypeAttribute.JumpNode](#), [NodeTypeAttribute.NormalNode](#), [NodeTypeAttribute.VariantNode](#),
[NodeTypeAttribute.EndNode](#) }

The nodes are sorted in their different node types

Öffentliche Methoden

- [UseCaseGraph](#) (params [IAttribute\[\]](#) attributes)
creates a new use case graph with the given attributes
- override string [ToString](#) ()
returns the use case graph as a string by returning its name attribute
- void [RecalculateScenarios](#) ()
sets the scenarios to null, so they will be initialized again when getting the property.

Statische öffentliche Attribute

- static readonly string[] [UseCaseGraphAttributeNames](#)
The expressions in the use case table
- static readonly string[] [NodeAttributeNames](#)
The attribute names of the graph nodes. You can access this array with the enum NodeAttributes

Propertys

- IEnumerable< IGraph > [Scenarios](#) [get]
scenarios of the use case graph lazy initialized when getter is called

Private Methoden

- void [InitAttribute](#)< T > ([UseCaseAttributes](#) attribute, T value)

Private Attribute

- IEnumerable< IGraph > [mScenarios](#)

6.10.1 Ausführliche Beschreibung

class to represent a use case.

6.10.2 Dokumentation der Aufzählungstypen

6.10.2.1 enum UseCaseAnalyser.Model.Model.UseCaseGraph.NodeTypeAttribute

The nodes are sorted in their different node types

Aufzählungswerte

- StartNode** The node with which the use case starts
- JumpNode** A variant sequence node which is connected with a normal routine node
- NormalNode** A normal routine node
- VariantNode** a variant node
- EndNode** a node which ends the use case

6.10.3 Beschreibung der Konstruktoren und Destruktoren

6.10.3.1 UseCaseAnalyser.Model.Model.UseCaseGraph.UseCaseGraph (params IAttribute[] attributes)

creates a new use case graph with the given attributes

Parameter

<i>attributes</i>	attributes to add to the use case graph
-------------------	---

6.10.4 Dokumentation der Elementfunktionen

6.10.4.1 `void UseCaseAnalyser.Model.Model.UseCaseGraph.InitAttribute< T > (UseCaseAttributes attribute, T value)`
[private]

6.10.4.2 `void UseCaseAnalyser.Model.Model.UseCaseGraph.RecalculateScenarios ()`

sets the scenarios to null, so they will be initialized again when getting the property.

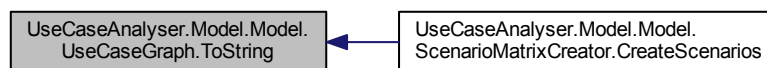
6.10.4.3 `override string UseCaseAnalyser.Model.Model.UseCaseGraph.ToString ()`

returns the use case graph as a string by returning its name attribute

Rückgabe

the use case graph as string

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



6.10.5 Dokumentation der Datenelemente

6.10.5.1 `IEnumerable< IGraph > UseCaseAnalyser.Model.Model.UseCaseGraph.mScenarios` [private]

6.10.5.2 `readonly string [] UseCaseAnalyser.Model.Model.UseCaseGraph.NodeAttributeNames` [static]

Initialisierung:

```

=
{
    "Normal Index",
    "Variant Index",
    "Variant Sequence Step",
    "Description",
    "NodeType"
}
  
```

The attribute names of the graph nodes. You can access this array with the enum NodeAttributes

6.10.5.3 `readonly string [] UseCaseAnalyser.Model.Model.UseCaseGraph.UseCaseGraphAttributeNames` [static]

Initialisierung:

```

=
{
    "Name",
    "Kennung",
    "Priorität",
    "Kurzbeschreibung:",
    "Vorbedingung(en) :",
    "Nachbedingung(en) :",
    "Normaler Ablauf:",
    "Ablauf-Varianten:",
}
  
```



```

        "Spezielle Anforderungen:",
        "Zu klärende Punkte:",
        "Varianten-Traversierungs-Anzahl",
        "Schleifen-Traversierungs-Anzahl"
    }

```

The expressions in the use case table

6.10.6 Dokumentation der Propertys

6.10.6.1 IEnumerable<IGraph> UseCaseAnalyser.Model.Model.UseCaseGraph.Scenarios [get]

scenarios of the use case graph lazy initialized when getter is called

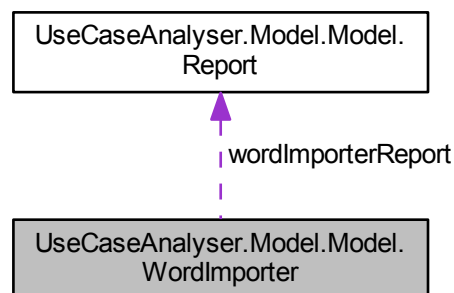
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Model/[UseCaseGraph.cs](#)

6.11 UseCaseAnalyser.Model.Model.WordImporter Klassenreferenz

Imports use case graphs from a word document

Zusammengehörigkeiten von UseCaseAnalyser.Model.Model.WordImporter:



Öffentliche, statische Methoden

- static List< [UseCaseGraph](#) > [ImportUseCases](#) (FileInfo file)
Imports all use cases that can be found in the file.
- static List< [UseCaseGraph](#) > [ImportUseCases](#) (FileInfo file, out [Report](#) report)
Imports all use cases that can be found in the file. It also generates a report für errors, warnings and log entries

Private, statische Methoden

- static bool [IsUseCaseTableFormat](#) ([Table](#) table)
Workaround: Checks if table fullfills format conditions to be a use case table. No content checks are performed - these are performed while parsing an table because format warning/errors are creted there. A better approach to this problem would be to parse all table and suppress warnings/errors if it is ensured that table is not an UseCase table. Because this functionality is not implemented yet, use this workaround to filter some table that are definitely no UC table only by its format. A valid use case table should be look like: +-----+

- static bool [TryGetUseCaseName](#) (TableRow row, [UseCaseGraph](#) useCaseGraph)
Tries to get the name of the use case which is always in the first cell in the first row in the use case table
- static bool [TryReadInUseCase](#) (IEnumerable< TableRow > tableRows, out [UseCaseGraph](#) useCaseGraph)
Tries to interpret the given table as a use case
- static bool [TryGetUseCaseld](#) (TableRow row, out string id)
Tries to get the use case id from the given row and checks if the id doesn't occurs twice
- static bool [TryGetHorizontalContent](#) (TableRow row, out string result, string heading)
Checks if the heading is correct, if the format is correct and returns the content string and true if successful or false if not
- static bool [TryGetVerticalContent](#) (IReadOnlyList< TableRow > rows, int actRowIndex, out string result, string heading)
This function tries to fetch the content from a use case table, that is declared underneath each other, e.g. "↵ Kurzbeschreibung", "Vorbereitung", etc.
- static bool [TryGetNormalRoutineAndSeqVars](#) ([UseCaseGraph](#) useCaseGraph, IReadOnlyList< TableRow > rows, int rowIndex)
Tries to get the normal routine (Normaler Ablauf) of the use case
- static WordprocessingDocument [FixedOpen](#) (string docPath, bool isEditable)
Workaround for OpenXML bug regarding Invalid hyperlinks exception (thrown by System.IO.Packaging) source: <http://openxmldeveloper.org/blog/b/openxmldeveloper/archive/2014/08/19/handling-invalid-hyperlinks.aspx> Check after first try of opening whether Invalid hyperlink exception was thrown. If it was thrown, create a copy of the document and remove invalid hyperlinks. Afterwards try to read content of fixed document. If there are still opening errors these exceptions will be forwarded otherwise a WordprocessingDocument reference will be returned.
- static Uri [FixUri](#) ()
Handler method for dealing with broken URIs.
- static void [FixInvalidUri](#) (Stream fs, Func< Uri > invalidUriHandler)
Replaces all invalid URIs from a stream by using invalidUriHandler and writes it back into the stream.

Private Attribute

- const string [UseCaseJump](#) = "Weiter mit:"
The expression which initiates a jump to another use case
- const string [SequenceJump](#) = "Rückkehr nach:"
The expression which initiates a jump from the sequence variant to the normal routine
- const string [UseCaseEnd](#) = "Ende."
The expression which defines the end of the use case

Statische, private Attribute

- static [Report](#) wordImporterReport
The report of the actual import process
- static string [actUseCaseld](#)
The use case ID of the actual use case
- static List< [UseCaseGraph](#) > [actUseCases](#)
List of all use case graphs in the actual import process

6.11.1 Ausführliche Beschreibung

Imports use case graphs from a word document

6.11.2 Dokumentation der Elementfunktionen

6.11.2.1 `static WordprocessingDocument UseCaseAnalyser.Model.Model.WordImporter.FixedOpen (string docPath, bool isEditable) [static],[private]`

Workaround for OpenXML bug regarding Invalid hyperlinks exception (thrown by System.IO.Packaging) source: <http://openxmldeveloper.org/blog/b/openxmldeveloper/archive/2014/08/19/handling-invalid.aspx> Check after first try of opening whether Invalid hyperlink exception was thrown. If it was thrown, create a copy of the document and remove invalid hyperlinks. Afterwards try to read content of fixed document. If there are still opening errors these exceptions will be forwarded otherwise a WordprocessingDocument reference will be returned.

Parameter

<i>docPath</i>	Path of docuemnt which should be opened.
<i>isEditable</i>	Flag for enable editing while open the file.

Rückgabe

Reference of opened WordprocessingDocument.

6.11.2.2 `static void UseCaseAnalyser.Model.Model.WordImporter.FixInvalidUri (Stream fs, Func< Uri > invalidUriHandler) [static],[private]`

Replaces all invalid URIs from a stream by using invalidUriHandler and writes it back into the stream.

Parameter

<i>fs</i>	Stream that should be checked and fixed regarding invalid URIs.
<i>invalidUriHandler</i>	Function that should be called if URI is invalid.

6.11.2.3 `static Uri UseCaseAnalyser.Model.Model.WordImporter.FixUri () [static],[private]`

Handler method for dealing with broken URIs.

Rückgabe

Returns a valid (dummy) URI.

6.11.2.4 `static List<UseCaseGraph> UseCaseAnalyser.Model.Model.WordImporter.ImportUseCases (FileInfo file) [static]`

Imports all use cases that can be found in the file.

Parameter

<i>file</i>	the word document (.docx)
-------------	---------------------------

Rückgabe

list of the use case graphs generated from the file

6.11.2.5 `static List<UseCaseGraph> UseCaseAnalyser.Model.Model.WordImporter.ImportUseCases (FileInfo file, out Report report) [static]`

Imports all use cases that can be found in the file. It also generates a report für errors, warnings and log entries

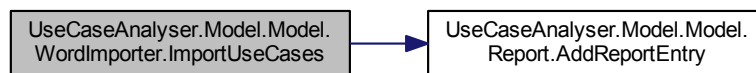
Parameter

<i>file</i>	the word document (.docx)
<i>report</i>	the report

Rückgabe

list of all use case graphs

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



6.11.2.6 `static bool UseCaseAnalyser.Model.Model.WordImporter.IsUseCaseTableFormat (Table table) [static], [private]`

Workaround: Checks if table fullfills format conditions to be a use case table. No content checks are performed - these are performed while parsing an table because format warning/errors are creted there. A better approach to this problem would be to parse all table and suppress warnings/errors if it is ensured that table is not an UseCase table. Because this functionality is not implemented yet, use this workaround to filter some table that are definitely no UC table only by its format. A valid use case table should be look like: +-----+

- (UC-Name) + +-----+
- ("Kennung") | (UC-Kennung) + +-----+
- ("Priority") | (UC-Priority) + +-----+
- ("Kurzbeschreibung") + +-----+
- (Short description) + +-----+
- ("Vorbedingung") + +-----+
- (preconditions) + +-----+
- ("Nachbedingung") + +-----+
- (postconditions) + +-----+
- ("Normaler Ablauf") + +-----+
- empty | (Normal sequence steps) + +-----+
- ("Ablauf-Varianten") + +-----+

OPTIONAL {

- Variant-Index | Variant trigger description + +-----+
- empty | (Variant sequence steps) + +-----+
- + ... + can be more more just one...

```
}
```

```
+-----+
```

- ("Spezielle Anforderungen") + +-----+
- (special conditions) + +-----+
- ("Zu klärende Punkte") + +-----+
- (open issues for clarification) + +-----+

Parameter

<i>table</i>	table that should be checked
--------------	------------------------------

Rückgabe

true if table fullfills use case table formats, false if not

6.11.2.7 `static bool UseCaseAnalyser.Model.Model.WordImporter.TryGetHorizontalContent (TableRow row, out string result, string heading) [static], [private]`

Checks if the heading is correct, if the format is correct and returns the content string and true if successful or false if not

Parameter

<i>row</i>	row where the content is expected
<i>result</i>	the cell content
<i>heading</i>	the heading

Rückgabe

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



6.11.2.8 `static bool UseCaseAnalyser.Model.Model.WordImporter.TryGetNormalRoutineAndSeqVars (UseCaseGraph useCaseGraph, IReadOnlyList< TableRow > rows, int rowIndex) [static], [private]`

Tries to get the normal routine (Normaler Ablauf) of the use case

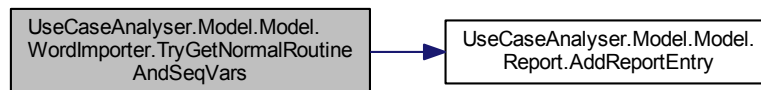
Parameter

<i>useCaseGraph</i>	
<i>rows</i>	
<i>rowIndex</i>	

Rückgabe

true when the use case could be successfully parsed

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



6.11.2.9 static bool UseCaseAnalyser.Model.Model.WordImporter.TryGetUseCaseId (TableRow row, out string id) [static],[private]

Tries to get the use case id from the given row and checks if the id doesn't occurs twice

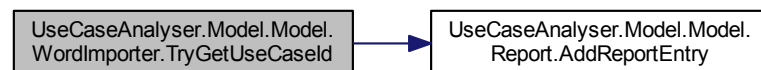
Parameter

<i>row</i>	the row with the id cell
<i>id</i>	the use case id

Rückgabe

true if success

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



6.11.2.10 static bool UseCaseAnalyser.Model.Model.WordImporter.TryGetUseCaseName (TableRow row, UseCaseGraph useCaseGraph) [static],[private]

Tries to get the name of the use case which is always in the first cell in the first row in the use case table

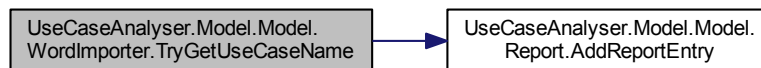
Parameter

<i>row</i>	the first row of the table
<i>useCaseGraph</i>	the actual use case graph

Rückgabe

true if success, false if not

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



6.11.2.11 `static bool UseCaseAnalyser.Model.Model.WordImporter.TryGetVerticalContent (IReadOnlyList< TableRow > rows, int actRowIndex, out string result, string heading) [static],[private]`

This function tries to fetch the content from a use case table, that is declared underneath each other, e.g. "↔ Kurzbeschreibung", "Vorbedingung", etc.

Parameter

<i>rows</i>	List of the table rows of this use case
<i>actRowIndex</i>	index of the row the heading is located
<i>result</i>	the cell content
<i>heading</i>	the heading

Rückgabe

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



6.11.2.12 `static bool UseCaseAnalyser.Model.Model.WordImporter.TryReadInUseCase (IEnumerable< TableRow > tableRows, out UseCaseGraph useCaseGraph) [static],[private]`

Tries to interpret the given table as a use case

Parameter

<i>tableRows</i>	
<i>useCaseGraph</i>	

Rückgabe

False when a error occurs

6.11.3 Dokumentation der Datenelemente

6.11.3.1 `string UseCaseAnalyser.Model.Model.WordImporter.actUseCaseId` [static],[private]

The use case ID of the actual use case

6.11.3.2 `List<UseCaseGraph> UseCaseAnalyser.Model.Model.WordImporter.actUseCases` [static],[private]

List of all use case graphs in the actual import process

6.11.3.3 `const string UseCaseAnalyser.Model.Model.WordImporter.SequenceJump = "Rückkehr nach:"` [private]

The expression which initiates a jump from the sequence variant to the normal routine

6.11.3.4 `const string UseCaseAnalyser.Model.Model.WordImporter.UseCaseEnd = "Ende."` [private]

The expression which defines the end of the use case

6.11.3.5 `const string UseCaseAnalyser.Model.Model.WordImporter.UseCaseJump = "Weiter mit:"` [private]

The expression which initiates a jump to another use case

6.11.3.6 `Report UseCaseAnalyser.Model.Model.WordImporter.wordImporterReport` [static],[private]

The report of the actual import process

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Model/WordImporter.cs](#)

Kapitel 7

Datei-Dokumentation

7.1 -global-/UseCaseAttributeExtensions.cs-Dateireferenz

Klassen

- class [UseCaseAttributeExtensions](#)
extension methods for easier attribute access

Typdefinitionen

- using [Attribute](#) = GraphFramework.Attribute

7.1.1 Dokumentation der benutzerdefinierten Typen

7.1.1.1 using [Attribute](#) = GraphFramework.Attribute

7.2 Model/HiddenAttribute.cs-Dateireferenz

Klassen

- class [UseCaseAnalyser.Model.Model.HiddenAttribute](#)
a marker class for attributes to filter some attributes from displaying in the view

Namensbereiche

- package [UseCaseAnalyser.Model.Model](#)

7.3 Model/Report.cs-Dateireferenz

Klassen

- class [UseCaseAnalyser.Model.Model.Report](#)
The report class
- class [UseCaseAnalyser.Model.Model.Report.ReportEntry](#)
Data holder

Namensbereiche

- package [UseCaseAnalyser.Model.Model](#)

7.4 Model/ScenarioMatrixCreator.cs-Dateireferenz

Klassen

- class [UseCaseAnalyser.Model.Model.ScenarioMatrixCreator](#)
class to create the scenarios for a use case graph

Namensbereiche

- package [UseCaseAnalyser.Model.Model](#)

Typdefinitionen

- using [Attribute](#) = GraphFramework.Attribute

7.4.1 Dokumentation der benutzerdefinierten Typen

7.4.1.1 using [Attribute](#) = GraphFramework.Attribute

7.5 Model/ScenarioMatrixExporter.cs-Dateireferenz

Klassen

- class [UseCaseAnalyser.Model.Model.ScenarioMatrixExporter](#)
class to export the scenario matrix to a file

Namensbereiche

- package [UseCaseAnalyser.Model.Model](#)

7.6 Model/UseCaseGraph.cs-Dateireferenz

Klassen

- class [UseCaseAnalyser.Model.Model.UseCaseGraph](#)
class to represent a use case.

Namensbereiche

- package [UseCaseAnalyser.Model.Model](#)

Aufzählungen

- enum [UseCaseAnalyser.Model.Model.UseCaseAttributes](#) {
[UseCaseAnalyser.Model.Model.UseCaseAttributes.Name](#) = 0, [UseCaseAnalyser.Model.Model.UseCaseAttributes.Id](#), [UseCaseAnalyser.Model.Model.UseCaseAttributes.Priority](#), [UseCaseAnalyser.Model.Model.UseCaseAttributes.Description](#),
[UseCaseAnalyser.Model.Model.UseCaseAttributes.PreCondition](#), [UseCaseAnalyser.Model.Model.UseCaseAttributes.PostCondition](#), [UseCaseAnalyser.Model.Model.UseCaseAttributes.NormalRoutine](#), [UseCaseAnalyser.Model.Model.UseCaseAttributes.SequenceVariation](#),
[UseCaseAnalyser.Model.Model.UseCaseAttributes.SpecialRequirements](#), [UseCaseAnalyser.Model.Model.UseCaseAttributes.OpenPoints](#), [UseCaseAnalyser.Model.Model.UseCaseAttributes.TraverseVariantCount](#), [UseCaseAnalyser.Model.Model.UseCaseAttributes.TraverseLoopCount](#) }

The access enum to the array UseCaseGraphAttributeNames

- enum [UseCaseAnalyser.Model.Model.NodeAttributes](#) {
[UseCaseAnalyser.Model.Model.NodeAttributes.NormalIndex](#), [UseCaseAnalyser.Model.Model.NodeAttributes.VariantIndex](#), [UseCaseAnalyser.Model.Model.NodeAttributes.VarSeqStep](#), [UseCaseAnalyser.Model.Model.NodeAttributes.Description](#),
[UseCaseAnalyser.Model.Model.NodeAttributes.NodeType](#) }

This enum is used to access the attribute names of the string array NodeAttributeNames

7.7 Model/WordImporter.cs-Dateireferenz

Klassen

- class [UseCaseAnalyser.Model.Model.WordImporter](#)
Imports use case graphs from a word document

Namensbereiche

- package [UseCaseAnalyser.Model.Model](#)

Typdefinitionen

- using [Attribute](#) = GraphFramework.Attribute
- using [Table](#) = DocumentFormat.OpenXml.Wordprocessing.Table

7.7.1 Dokumentation der benutzerdefinierten Typen

7.7.1.1 using [Attribute](#) = GraphFramework.Attribute

7.7.1.2 using [Table](#) = DocumentFormat.OpenXml.Wordprocessing.Table

7.8 Properties/AssemblyInfo.cs-Dateireferenz

7.9 ViewModel/AsyncCommand.cs-Dateireferenz

Klassen

- class [UseCaseAnalyser.Model.ViewModel.AsyncCommand](#)
implementation of the icommand interface. used to bind to from view side

Namensbereiche

- package [UseCaseAnalyser.Model.ViewModel](#)

7.10 ViewModel/DialogViewModel.cs-Dateireferenz

Klassen

- class [UseCaseAnalyser.Model.ViewModel.DialogViewModel](#)
main view model of the application provides all properties which will be displayed in view

Namensbereiche

- package [UseCaseAnalyser.Model.ViewModel](#)

7.11 ViewModel/IDialogView.cs-Dateireferenz

Klassen

- interface [UseCaseAnalyser.Model.ViewModel.IDialogView](#)
abstraction of the dialog view used to execute view actions from viewmodel side

Namensbereiche

- package [UseCaseAnalyser.Model.ViewModel](#)

Aufzählungen

- enum [UseCaseAnalyser.Model.ViewModel.MessageType](#) { [UseCaseAnalyser.Model.ViewModel.MessageType.Information](#), [UseCaseAnalyser.Model.ViewModel.MessageType.Warning](#), [UseCaseAnalyser.Model.ViewModel.MessageType.Error](#) }
enum for the different message types to be displayed in message boxes
- enum [UseCaseAnalyser.Model.ViewModel.FileDialogType](#) { [UseCaseAnalyser.Model.ViewModel.FileDialogType.Open](#), [UseCaseAnalyser.Model.ViewModel.FileDialogType.Save](#) }
enum for the different file dialog types

Index

-global-/UseCaseAttributeExtensions.cs, [49](#)

actUseCaseId
 UseCaseAnalyser::Model::Model::WordImporter, [48](#)

actUseCases
 UseCaseAnalyser::Model::Model::WordImporter, [48](#)

AddReportEntry
 UseCaseAnalyser::Model::Model::Report, [24](#)

AsyncCommand
 UseCaseAnalyser::Model::ViewModel::Async↔
 Command, [14](#)

Attribute
 ScenarioMatrixCreator.cs, [50](#)
 UseCaseAttributeExtensions, [33](#), [34](#)
 UseCaseAttributeExtensions.cs, [49](#)
 WordImporter.cs, [51](#)

AttributeName
 UseCaseAttributeExtensions, [34](#)

AttributeValue< T >
 UseCaseAttributeExtensions, [35](#)

ByName
 UseCaseAttributeExtensions, [35](#)

COrder
 UseCaseAnalyser::Model::Model::Scenario↔
 MatrixCreator, [30](#)
 UseCaseAnalyser::Model::Model::Scenario↔
 MatrixExporter, [32](#)

CScenarioName
 UseCaseAnalyser::Model::Model::Scenario↔
 MatrixCreator, [30](#)

CUseCase
 UseCaseAnalyser::Model::Model::Scenario↔
 MatrixCreator, [30](#)

CanExecute
 UseCaseAnalyser::Model::ViewModel::Async↔
 Command, [14](#)

CanExecuteChanged
 UseCaseAnalyser::Model::ViewModel::Async↔
 Command, [15](#)

Content
 UseCaseAnalyser::Model::Model::Report::↔
 ReportEntry, [28](#)

CountVariants
 UseCaseAnalyser::Model::Model::Scenario↔
 MatrixCreator, [29](#)

CreateAndFillExcelPages
 UseCaseAnalyser::Model::Model::Scenario↔
 MatrixExporter, [31](#)

CreateAttribute< T >
 UseCaseAttributeExtensions, [35](#), [37](#)

CreateAttribute< TValue >
 UseCaseAttributeExtensions, [37](#)

CreateScenarioMatrix
 UseCaseAnalyser::Model::Model::Scenario↔
 MatrixCreator, [29](#)

CreateScenarios
 UseCaseAnalyser::Model::Model::Scenario↔
 MatrixCreator, [29](#)

DEFAULT
 UseCaseAnalyser::Model::Model::Report, [24](#)

Description
 UseCaseAnalyser::Model::Model, [10](#)

DialogViewModel
 UseCaseAnalyser::Model::ViewModel::Dialog↔
 ViewModel, [17](#)

ERROR
 UseCaseAnalyser::Model::Model::Report, [24](#)

EndNode
 UseCaseAnalyser::Model::Model::UseCaseGraph, [39](#)

Entrytype
 UseCaseAnalyser::Model::Model::Report, [24](#)

Error
 UseCaseAnalyser::Model::ViewModel, [11](#)

ErrorReportEntries
 UseCaseAnalyser::Model::Model::Report, [26](#)

ExcelExtension
 UseCaseAnalyser::Model::Model::Scenario↔
 MatrixExporter, [32](#)

Execute
 UseCaseAnalyser::Model::ViewModel::Async↔
 Command, [15](#)

ExportAllScenarioMatrices
 UseCaseAnalyser::Model::ViewModel::Dialog↔
 ViewModel, [18](#)

ExportScenarioMatrix
 UseCaseAnalyser::Model::Model::Scenario↔
 MatrixExporter, [32](#)
 UseCaseAnalyser::Model::ViewModel::Dialog↔
 ViewModel, [18](#)

ExtendOrderAttribute
 UseCaseAnalyser::Model::Model::Scenario↔
 MatrixCreator, [29](#)

- FileDialogType
 - UseCaseAnalyser::Model::ViewModel, 11
- FindStartNode
 - UseCaseAnalyser::Model::Model::Scenario↔
 - MatrixCreator, 29
- FixInvalidUri
 - UseCaseAnalyser::Model::Model::WordImporter, 43
- FixUri
 - UseCaseAnalyser::Model::Model::WordImporter, 43
- FixedOpen
 - UseCaseAnalyser::Model::Model::WordImporter, 43
- GetEntriesByTag
 - UseCaseAnalyser::Model::Model::Report, 25
- GetExcelAddressFromXY
 - UseCaseAnalyser::Model::Model::Scenario↔
 - MatrixExporter, 32
- GetNodeNumber
 - UseCaseAnalyser::Model::Model::Scenario↔
 - MatrixCreator, 29
- Heading
 - UseCaseAnalyser::Model::Model::Report::↔
 - ReportEntry, 28
- HiddenAttribute
 - UseCaseAnalyser::Model::Model::HiddenAttribute, 20
- Id
 - UseCaseAnalyser::Model::Model, 10
- ImportUseCases
 - UseCaseAnalyser::Model::Model::WordImporter, 43
- Information
 - UseCaseAnalyser::Model::ViewModel, 11
- InitAttribute< T >
 - UseCaseAnalyser::Model::Model::UseCaseGraph, 40
- IsAlternativeNode
 - UseCaseAnalyser::Model::Model::Scenario↔
 - MatrixCreator, 29
- IsEndNode
 - UseCaseAnalyser::Model::Model::Scenario↔
 - MatrixCreator, 29
- IsUseCaseTableFormat
 - UseCaseAnalyser::Model::Model::WordImporter, 44
- IsVariantEntry
 - UseCaseAnalyser::Model::Model::Scenario↔
 - MatrixCreator, 30
- JumpNode
 - UseCaseAnalyser::Model::Model::UseCaseGraph, 39
- LOG
 - UseCaseAnalyser::Model::Model::Report, 24
- LatestWordImportReport
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
 - ViewModel, 18
- LogReportEntries
 - UseCaseAnalyser::Model::Model::Report, 26
- mCanExecuteFunc
 - UseCaseAnalyser::Model::ViewModel::Async↔
 - Command, 15
- mErrorReportEntries
 - UseCaseAnalyser::Model::Model::Report, 25
- mExecuteAction
 - UseCaseAnalyser::Model::ViewModel::Async↔
 - Command, 15
- mExportAllScenarioMatrices
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
 - ViewModel, 18
- mExportScenarioMatrix
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
 - ViewModel, 18
- mIsExecuting
 - UseCaseAnalyser::Model::ViewModel::Async↔
 - Command, 15
- mLogReportEntries
 - UseCaseAnalyser::Model::Model::Report, 25
- mOnError
 - UseCaseAnalyser::Model::ViewModel::Async↔
 - Command, 15
- mOpenAboutView
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
 - ViewModel, 18
- mOpenLogfile
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
 - ViewModel, 18
- mOpenReportView
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
 - ViewModel, 18
- mOpenWordFile
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
 - ViewModel, 18
- mRefreshGraph
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
 - ViewModel, 18
- mScenarios
 - UseCaseAnalyser::Model::Model::UseCaseGraph, 40
- mSelectedScenario
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
 - ViewModel, 18
- mSelectedUseCaseGraph
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
 - ViewModel, 18
- mUseCaseGraphs
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
 - ViewModel, 18
- mViewAbstraction
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
 - ViewModel, 18

- mWarningReportEntries
 - UseCaseAnalyser::Model::Model::Report, [25](#)
- MessageType
 - UseCaseAnalyser::Model::ViewModel, [11](#)
- Model/HiddenAttribute.cs, [49](#)
- Model/Report.cs, [49](#)
- Model/ScenarioMatrixCreator.cs, [50](#)
- Model/ScenarioMatrixExporter.cs, [50](#)
- Model/UseCaseGraph.cs, [50](#)
- Model/WordImporter.cs, [51](#)
- Name
 - UseCaseAnalyser::Model::Model, [10](#)
- NodeAttributeNames
 - UseCaseAnalyser::Model::Model::UseCaseGraph, [40](#)
- NodeAttributes
 - UseCaseAnalyser::Model::Model, [10](#)
- NodesNodeInBranch
 - UseCaseAnalyser::Model::Model::Scenario↔
MatrixExporter, [32](#)
- NodeType
 - UseCaseAnalyser::Model::Model, [10](#)
- NodeTypeAttribute
 - UseCaseAnalyser::Model::Model::UseCaseGraph, [39](#)
- NormalIndex
 - UseCaseAnalyser::Model::Model, [10](#)
- NormalNode
 - UseCaseAnalyser::Model::Model::UseCaseGraph, [39](#)
- NormalRoutine
 - UseCaseAnalyser::Model::Model, [10](#)
- OnError
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
ViewModel, [18](#)
- OnPropertyChanged
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
ViewModel, [18](#)
- Open
 - UseCaseAnalyser::Model::ViewModel, [11](#)
- OpenAboutView
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
ViewModel, [19](#)
 - UseCaseAnalyser::Model::ViewModel::IDialog↔
View, [21](#)
- OpenFileDialog
 - UseCaseAnalyser::Model::ViewModel::IDialog↔
View, [21](#)
- OpenLogfile
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
ViewModel, [19](#)
- OpenMessageBox
 - UseCaseAnalyser::Model::ViewModel::IDialog↔
View, [21](#)
- OpenPoints
 - UseCaseAnalyser::Model::Model, [10](#)
- OpenReportResult
 - UseCaseAnalyser::Model::ViewModel::IDialog↔
View, [23](#)
- OpenReportView
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
ViewModel, [19](#)
- OpenWordFile
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
ViewModel, [19](#)
- PostCondition
 - UseCaseAnalyser::Model::Model, [10](#)
- PreCondition
 - UseCaseAnalyser::Model::Model, [10](#)
- Priority
 - UseCaseAnalyser::Model::Model, [10](#)
- Properties/AssemblyInfo.cs, [51](#)
- PropertyChanged
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
ViewModel, [19](#)
- RecalculateScenarios
 - UseCaseAnalyser::Model::Model::UseCaseGraph, [40](#)
- RedrawGraph
 - UseCaseAnalyser::Model::ViewModel::IDialog↔
View, [23](#)
- RefreshGraph
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
ViewModel, [19](#)
- ReportEntry
 - UseCaseAnalyser::Model::Model::Report::↔
ReportEntry, [27](#)
- Save
 - UseCaseAnalyser::Model::ViewModel, [11](#)
- ScenarioMatrixCreator.cs
 - Attribute, [50](#)
- Scenarios
 - UseCaseAnalyser::Model::Model::UseCaseGraph, [41](#)
- SelectedScenario
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
ViewModel, [19](#)
- SelectedUseCaseGraph
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
ViewModel, [19](#)
- SequenceJump
 - UseCaseAnalyser::Model::Model::WordImporter, [48](#)
- SequenceVariation
 - UseCaseAnalyser::Model::Model, [10](#)
- SpecialRequirements
 - UseCaseAnalyser::Model::Model, [10](#)
- StartNode
 - UseCaseAnalyser::Model::Model::UseCaseGraph, [39](#)
- Table
 - WordImporter.cs, [51](#)

- Tag
 - UseCaseAnalyser::Model::Model::Report::↔
ReportEntry, 28
- ToString
 - UseCaseAnalyser::Model::Model::UseCaseGraph, 40
- TraverseLoopCount
 - UseCaseAnalyser::Model::Model, 10
- TraverseVariantCount
 - UseCaseAnalyser::Model::Model, 10
- TryGetHorizontalContent
 - UseCaseAnalyser::Model::Model::WordImporter, 45
- TryGetNormalRoutineAndSeqVars
 - UseCaseAnalyser::Model::Model::WordImporter, 45
- TryGetUseCaseld
 - UseCaseAnalyser::Model::Model::WordImporter, 46
- TryGetUseCaseName
 - UseCaseAnalyser::Model::Model::WordImporter, 46
- TryGetVerticalContent
 - UseCaseAnalyser::Model::Model::WordImporter, 47
- TryReadInUseCase
 - UseCaseAnalyser::Model::Model::WordImporter, 47
- Type
 - UseCaseAnalyser::Model::Model::Report::↔
ReportEntry, 28
- UiTaskFactory
 - UseCaseAnalyser::Model::ViewModel::Async↔
Command, 15
- UseCaseAnalyser, 9
- UseCaseAnalyser.Model, 9
- UseCaseAnalyser.Model.Model, 9
- UseCaseAnalyser.Model.Model.HiddenAttribute, 20
- UseCaseAnalyser.Model.Model.Report, 23
- UseCaseAnalyser.Model.Model.Report.ReportEntry, 26
- UseCaseAnalyser.Model.Model.ScenarioMatrixCreator, 28
- UseCaseAnalyser.Model.Model.ScenarioMatrix↔
Exporter, 30
- UseCaseAnalyser.Model.Model.UseCaseGraph, 37
- UseCaseAnalyser.Model.Model.WordImporter, 41
- UseCaseAnalyser.Model.ViewModel, 11
- UseCaseAnalyser.Model.ViewModel.AsyncCommand, 13
- UseCaseAnalyser.Model.ViewModel.DialogViewModel, 15
- UseCaseAnalyser.Model.ViewModel.IDialogView, 21
- UseCaseAnalyser::Model::Model
 - Description, 10
 - Id, 10
 - Name, 10
 - NodeAttributes, 10
 - NodeType, 10
 - NormalIndex, 10
 - NormalRoutine, 10
 - OpenPoints, 10
 - PostCondition, 10
 - PreCondition, 10
 - Priority, 10
 - SequenceVariation, 10
 - SpecialRequirements, 10
 - TraverseLoopCount, 10
 - TraverseVariantCount, 10
 - UseCaseAttributes, 10
 - VarSeqStep, 10
 - VariantIndex, 10
- UseCaseAnalyser::Model::Model::HiddenAttribute
 - HiddenAttribute, 20
- UseCaseAnalyser::Model::Model::Report
 - AddReportEntry, 24
 - DEFAULT, 24
 - ERROR, 24
 - Entrytype, 24
 - ErrorReportEntries, 26
 - GetEntriesByTag, 25
 - LOG, 24
 - LogReportEntries, 26
 - mErrorReportEntries, 25
 - mLogReportEntries, 25
 - mWarningReportEntries, 25
 - WARNING, 24
 - WarningReportEntries, 26
- UseCaseAnalyser::Model::Model::Report::ReportEntry
 - Content, 28
 - Heading, 28
 - ReportEntry, 27
 - Tag, 28
 - Type, 28
- UseCaseAnalyser::Model::Model::ScenarioMatrix↔
Creator
 - COrder, 30
 - CScenarioName, 30
 - CUseCase, 30
 - CountVariants, 29
 - CreateScenarioMatrix, 29
 - CreateScenarios, 29
 - ExtendOrderAttribute, 29
 - FindStartNode, 29
 - GetNodeNumber, 29
 - IsAlternativeNode, 29
 - IsEndNode, 29
 - IsVariantEntry, 30
- UseCaseAnalyser::Model::Model::ScenarioMatrix↔
Exporter
 - COrder, 32
 - CreateAndFillExcelPages, 31
 - ExcelExtension, 32
 - ExportScenarioMatrix, 32
 - GetExcelAdressFromXY, 32
 - NodeIsNodeInBranch, 32
 - ValidateFile, 32

- WriteScenarios, 32
- UseCaseAnalyser::Model::Model::UseCaseGraph
 - EndNode, 39
 - InitAttribute< T >, 40
 - JumpNode, 39
 - mScenarios, 40
 - NodeAttributeNames, 40
 - NodeTypeAttribute, 39
 - NormalNode, 39
 - RecalculateScenarios, 40
 - Scenarios, 41
 - StartNode, 39
 - ToString, 40
 - UseCaseGraph, 39
 - UseCaseGraphAttributeNames, 40
 - VariantNode, 39
- UseCaseAnalyser::Model::Model::WordImporter
 - actUseCaseId, 48
 - actUseCases, 48
 - FixInvalidUri, 43
 - FixUri, 43
 - FixedOpen, 43
 - ImportUseCases, 43
 - IsUseCaseTableFormat, 44
 - SequenceJump, 48
 - TryGetHorizontalContent, 45
 - TryGetNormalRoutineAndSeqVars, 45
 - TryGetUseCaseId, 46
 - TryGetUseCaseName, 46
 - TryGetVerticalContent, 47
 - TryReadInUseCase, 47
 - UseCaseEnd, 48
 - UseCaseJump, 48
 - wordImporterReport, 48
- UseCaseAnalyser::Model::ViewModel
 - Error, 11
 - FileDialogType, 11
 - Information, 11
 - MessageType, 11
 - Open, 11
 - Save, 11
 - Warning, 11
- UseCaseAnalyser::Model::ViewModel::AsyncCommand
 - AsyncCommand, 14
 - CanExecute, 14
 - CanExecuteChanged, 15
 - Execute, 15
 - mCanExecuteFunc, 15
 - mExecuteAction, 15
 - mIsExecuting, 15
 - mOnError, 15
 - UiTaskFactory, 15
- UseCaseAnalyser::Model::ViewModel::DialogView↔
 - Model
 - DialogViewModel, 17
 - ExportAllScenarioMatrices, 18
 - ExportScenarioMatrix, 18
 - LatestWordImportReport, 18
 - mExportAllScenarioMatrices, 18
 - mExportScenarioMatrix, 18
 - mOpenAboutView, 18
 - mOpenLogfile, 18
 - mOpenReportView, 18
 - mOpenWordFile, 18
 - mRefreshGraph, 18
 - mSelectedScenario, 18
 - mSelectedUseCaseGraph, 18
 - mUseCaseGraphs, 18
 - mViewAbstraction, 18
 - OnError, 18
 - OnPropertyChanged, 18
 - OpenAboutView, 19
 - OpenLogfile, 19
 - OpenReportView, 19
 - OpenWordFile, 19
 - PropertyChanged, 19
 - RefreshGraph, 19
 - SelectedScenario, 19
 - SelectedUseCaseGraph, 19
 - UseCaseGraphs, 19
- UseCaseAnalyser::Model::ViewModel::IDialogView
 - OpenAboutView, 21
 - OpenFileDialog, 21
 - OpenMessageBox, 21
 - OpenReportResult, 23
 - RedrawGraph, 23
- UseCaseAttributeExtensions, 33
 - Attribute, 33, 34
 - AttributeName, 34
 - AttributeValue< T >, 35
 - ByName, 35
 - CreateAttribute< T >, 35, 37
 - CreateAttribute< TValue >, 37
- UseCaseAttributeExtensions.cs
 - Attribute, 49
- UseCaseAttributes
 - UseCaseAnalyser::Model::Model, 10
- UseCaseEnd
 - UseCaseAnalyser::Model::Model::WordImporter, 48
- UseCaseGraph
 - UseCaseAnalyser::Model::Model::UseCaseGraph, 39
- UseCaseGraphAttributeNames
 - UseCaseAnalyser::Model::Model::UseCaseGraph, 40
- UseCaseGraphs
 - UseCaseAnalyser::Model::ViewModel::Dialog↔
 - ViewModel, 19
- UseCaseJump
 - UseCaseAnalyser::Model::Model::WordImporter, 48
- ValidateFile
 - UseCaseAnalyser::Model::Model::Scenario↔
 - MatrixExporter, 32
- VarSeqStep

- UseCaseAnalyser::Model::Model, [10](#)
- VariantIndex
 - UseCaseAnalyser::Model::Model, [10](#)
- VariantNode
 - UseCaseAnalyser::Model::Model::UseCaseGraph,
[39](#)
- ViewModel/AsyncCommand.cs, [51](#)
- ViewModel/DialogViewModel.cs, [52](#)
- ViewModel/IDialogView.cs, [52](#)
- WARNING
 - UseCaseAnalyser::Model::Model::Report, [24](#)
- Warning
 - UseCaseAnalyser::Model::ViewModel, [11](#)
- WarningReportEntries
 - UseCaseAnalyser::Model::Model::Report, [26](#)
- WordImporter.cs
 - Attribute, [51](#)
 - Table, [51](#)
- wordImporterReport
 - UseCaseAnalyser::Model::Model::WordImporter,
[48](#)
- WriteScenarios
 - UseCaseAnalyser::Model::Model::Scenario↔
MatrixExporter, [32](#)