UseCaseAnalyser.Model

Erzeugt von Doxygen 1.8.9.1

Die Jun 30 2015 08:47:51

Inhaltsverzeichnis

Kapitel 1

Verzeichnis der Namensbereiche

1.1 Pakete

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

Kapitel 2

Hierarchie-Verzeichnis

2.1 Klassenhierarchie

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

Kapitel 3

Klassen-Verzeichnis

3.1 Auflistung der Klassen

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

Kapitel 4

Datei-Verzeichnis

4.1 Auflistung der Dateien

Hier folgt die Aufzählung aller Dateien mit einer Kurzbeschreibung:

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

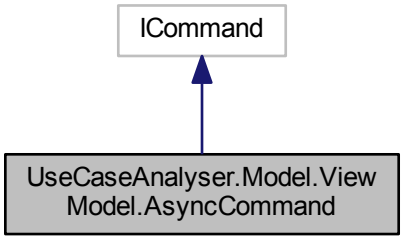
Kapitel 6

Klassen-Dokumentation

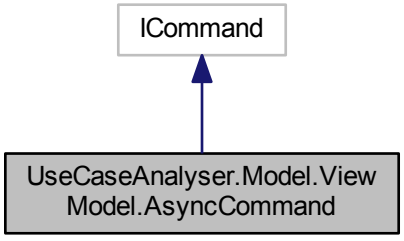
6.1 UseCaseAnalyser.Model.ViewModel.AsyncCommand Klassenreferenz

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

Propertys

- EventHandler CanExecuteChanged

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

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

executeAction	action to execute on command execute
canExecuteFunc	function to determine weather the action can be executed
onError	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

parameter	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

parameter	action parameter which can be passed from the view

6.1.4 Dokumentation der Propertys

6.1.4.1 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

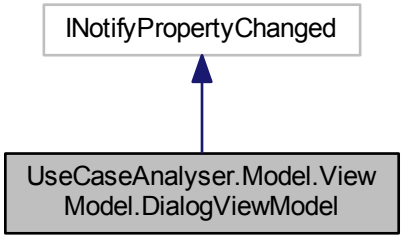
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- ViewModel/AsyncCommand.cs

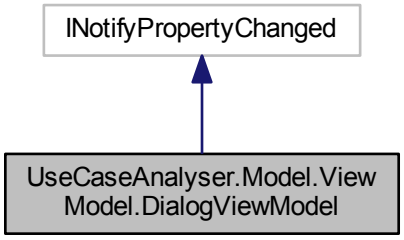
6.2 UseCaseAnalyser.Model.ViewModel.DialogViewModel Klassenreferenz

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]`
  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]`
  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

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

viewAbstraction	interface abstraction of the view

6.2.3 Dokumentation der Elementfunktionen

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

fires the property changed for the given property name

Parameter

propertyName	

6.2.4 Dokumentation der Propertys

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

exports the scenarios from all use cases

enabled if: any use case is imported

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

exports the scenarios from the currently selected use case

enabled if: a use case is selected

6.2.4.3 Report UseCaseAnalyser.Model.ViewModel.DialogViewModel.LatestWordImportReport `[get]`

the latest word import report gotten from the word importer

6.2.4.4 ICommand UseCaseAnalyser.Model.ViewModel.DialogViewModel.OpenAboutView `[get]`

opens the about view in its seperate window

6.2.4.5 ICommand UseCaseAnalyser.Model.ViewModel.DialogViewModel.OpenLogfile `[get]`

opens the logfile as seperate process

enabled if: the logfile exists

6.2.4.6 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

6.2.4.7 ICommand UseCaseAnalyser.Model.ViewModel.DialogViewModel.OpenWordFile `[get]`

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

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

refreshes the current use case graph visualization

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

the currently selection scenario from the view –> set via binding

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

the currently selected graph from the view –> set via binding

6.2.4.11 IEnumerable<UseCaseGraph> UseCaseAnalyser.Model.ViewModel.DialogViewModel.UseCaseGraphs [get]

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

6.2.5 Ereignisdokumentation

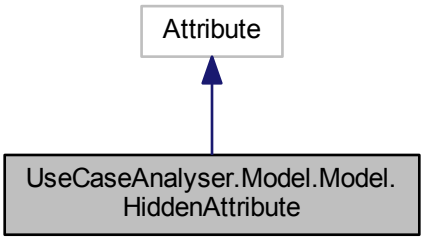
6.2.5.1 PropertyChangedEventHandler UseCaseAnalyser.Model.ViewModel.DialogViewModel.PropertyChanged

invoked to notify the gui about changed of properties
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

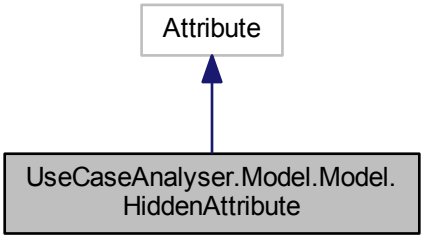
- ViewModel/DialogViewModel.cs

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

name	the name of the attribute
value	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

filter	filter of the files
dialogType	dialog type (open or save)
predefinedName	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

header	header of the message box
content	content of the message box
messageType	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

viewModel	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

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

entry	

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

tag	
type	optional

Rückgabe

6.5.4 Dokumentation der Propertys

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

All error report entries

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

All log report entries

6.5.4.3 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]

  A brief summary of the report entry
- string Content [get]

  A description of the report entry
- Entrytype Type [get]

  The type, e.g. log, error, warning
- string Tag [get]

  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

heading	heading of the report entry
content	content of the report entry
type	entry type (error, warning, log)
tag	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]

A description of the report entry

6.6.3.2 string UseCaseAnalyser.Model.Model.Report.ReportEntry.Heading [get]

A brief summary of the report entry

6.6.3.3 string UseCaseAnalyser.Model.Model.Report.ReportEntry.Tag [get]

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]

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.

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

Creates all scenarios from a Use-Case graph.

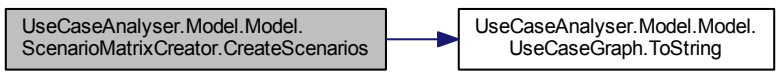
Parameter

useCaseGraph	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:



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 specfified file (.xlsx)

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.ExportScenarioMatrix (IEnumerable< UseCaseGraph > *useCaseGraphs,* FileInfo *file*) [static]

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

Parameter

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

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 attributeValue, 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

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 throwException is true and the attribute is not contained in the source item's attributes

Parameter

source	the node which contains the attributes
usecaseAttribute	the attribute type which should be read
throwException	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

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

source	the node which contains the attributes
nodeAttribute	the attribute type which should be read
throwException	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

sourceUse↩ CaseAttribute	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

sourceNode↩ Attribute	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

source	the node which contains the attributes
useCase↩ Attribute	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

source	the node which contains the attributes
nodeAttribute	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

sourceEnumerable	the enumerable which should contain the attribute with the given name
name	name of the attribute to search
throwException	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

sourceUsecasegraphAttribute	the nodeAttribute enum value
attributeValue	the value of the attribute
hidden	value to determine weather the attribute should be hidden in the gui

Rückgabe

a new attribute

6.9.2.9 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

sourceNodeAttribute	the nodeAttribute enum value
attributeValue	the value of the attribute
hidden	value to determine weather the attribute should be hidden in the gui

Rückgabe

a new attribute

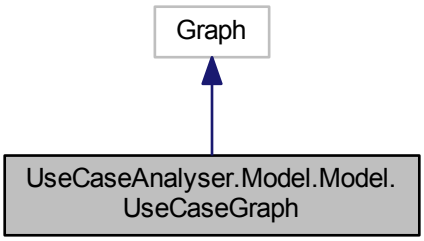
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- -global-/UseCaseAttributeExtensions.cs

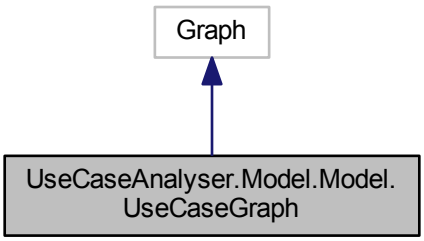
6.10 UseCaseAnalyser.Model.Model.UseCaseGraph Klassenreferenz

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, NodeType↩
  Attribute.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

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

attributes	attributes to add to the use case graph

6.10.4 Dokumentation der Elementfunktionen

6.10.4.1 void UseCaseAnalyser.Model.Model.UseCaseGraph.RecalculateScenarios ( )

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

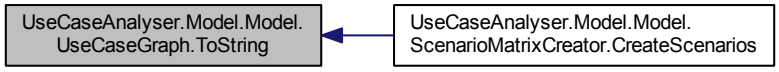
6.10.4.2 override string UseCaseAnalyser.Model.Model.UseCaseGraph.ToString ( )

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 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

6.10.5.2 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

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

6.11.1 Ausführliche Beschreibung

Imports use case graphs from a word document

6.11.2 Dokumentation der Elementfunktionen

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

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

Parameter

file	the word document (.docx)

Rückgabe

list of the use case graphs generated from the file

6.11.2.2 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

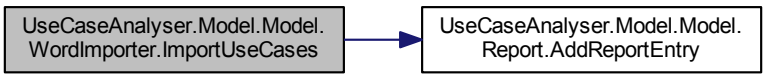
Parameter

file	the word document (.docx)
report	the report

Rückgabe

list of all use case graphs

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



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