NSA.Model

Erzeugt von Doxygen 1.8.11

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 NSA-Namensbereichsreferenz

Namensbereiche

- namespace Model

5.2 NSA.Model-Namensbereichsreferenz

Namensbereiche

- namespace BusinessLogic
- namespace NetworkComponents

5.3 NSA.Model.BusinessLogic-Namensbereichsreferenz

Namensbereiche

- namespace TestscenarioRunnables

Klassen

- class Packet

  Class for a packet
- class Project

  Class for project
- class Rule

  Rule class parses the text and if it is valid, the Rule object will be used for simulations
- class Simulation

  Class for the simulation
- class Testscenario

Aufzählungen

- enum SimulationType { SimulationType.Simple = 0, SimulationType.Only = 1, SimulationType.HasInternet = 2 }

  Simulation types

5.3.1 Dokumentation der Aufzählungstypen

5.3.1.1 enum NSA.Model.BusinessLogic.SimulationType [strong]

Simulation types

Aufzählungswerte

Simple
Only
HasInternet

5.4 NSA.Model.BusinessLogic.TestscenarioRunnables-Namensbereichsreferenz

Klassen

- class HasInternetTestscenarioRunnable
- interface ITestscenarioRunnable
- class OnlyTestscenarioRunnable
- class SimpleTestscenarioRunnable

5.5 NSA.Model.NetworkComponents-Namensbereichsreferenz

Namensbereiche

- namespace Helper_Classes
- namespace Layers

Klassen

- class Connection

  Class for a connection between two hardwarenodes.
- class Hardwarenode

  Implements the basis class for hardwarenodes.
- interface ILayer
- class Interface

  Class for a interface of an hardwarenode.
- class Layerstack

  Layerstack
- class Network
- class Route

  class for a single route of the routingtable
- class Router

  Implements the network component Router.
- class Switch

  Implements the network component switch.
- class Workstation

  Implements the network component Workstation.

5.6 NSA.Model.NetworkComponents.Helper_Classes-Namensbereichsreferenz

Klassen

- class IPAddressExtensions
- class Result

  Result class for the packetresult
- class ValidationInfo

  Helper class for the parameters of the simulation

5.7 NSA.Model.NetworkComponents.Layers-Namensbereichsreferenz

Klassen

- class ApplicationLayer

  Application-Layer
- class CustomLayer

  Custom-Layer
- class DataLinkLayer

  DataLink-Layer
- class NetworkLayer

  Network-Layer
- class PhysicalLayer

  Physical-Layer
- class PresentationLayer

  Presentation-Layer
- class SessionLayer

  Session-Layer
- class TransportLayer

  Transport-Layer

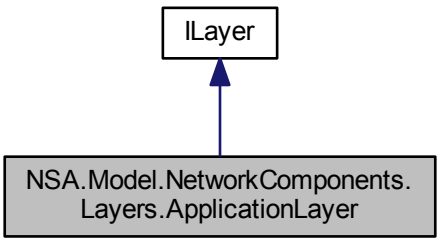
Kapitel 6

Klassen-Dokumentation

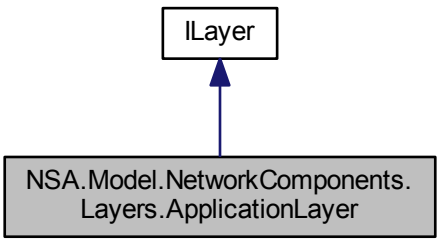
6.1 NSA.Model.NetworkComponents.Layers.ApplicationLayer Klassenreferenz

Application-Layer

Klassendiagramm für NSA.Model.NetworkComponents.Layers.ApplicationLayer:



Zusammengehörigkeiten von NSA.Model.NetworkComponents.Layers.ApplicationLayer:



Öffentliche Methoden

- ApplicationLayer (int I)

  Initializes a new instance of the ApplicationLayer class.
- bool ValidateReceive (Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)

  Validates the layer while receiving a packet.
- string GetLayerName ()

  Gets the name of the layer.
- bool SetLayerName (string NewName)

  Sets the name of the layer.
- int GetLayerIndex ()

  Gets the index of the layer.
- void SetLayerIndex (int I)

  Sets the index of the layer.
- void ValidateSend (Workstation Destination, Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, int LayerIndex)

  Validates the layer while sending a packet.

6.1.1 Ausführliche Beschreibung

Application-Layer

Siehe auch

NSA.Model.NetworkComponents.ILayer

6.1.2 Beschreibung der Konstruktoren und Destruktoren

6.1.2.1 NSA.Model.NetworkComponents.Layers.ApplicationLayer.ApplicationLayer (int *I*)

Initializes a new instance of the ApplicationLayer class.

Parameter

I	The index.

6.1.3 Dokumentation der Elementfunktionen

6.1.3.1 int NSA.Model.NetworkComponents.Layers.ApplicationLayer.GetLayerIndex ( )

Gets the index of the layer.

Rückgabe

The index

Implementiert NSA.Model.NetworkComponents.ILayer.

6.1.3.2 string NSA.Model.NetworkComponents.Layers.ApplicationLayer.GetLayerName ( )

Gets the name of the layer.

Rückgabe

The Layername

Implementiert NSA.Model.NetworkComponents.ILayer.

6.1.3.3 void NSA.Model.NetworkComponents.Layers.ApplicationLayer.SetLayerIndex (int *I*)

Sets the index of the layer.

Parameter

I	The Index.

Implementiert NSA.Model.NetworkComponents.ILayer.

6.1.3.4 bool NSA.Model.NetworkComponents.Layers.ApplicationLayer.SetLayerName (string *NewName*)

Sets the name of the layer.

Parameter

NewName	The new name

Rückgabe

Implementiert NSA.Model.NetworkComponents.ILayer.

6.1.3.5 bool NSA.Model.NetworkComponents.Layers.ApplicationLayer.ValidateReceive (Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* Hardwarenode *Destination,* int *LayerIndex*)

Validates the layer while receiving a packet.

Parameter

CurrentNode	Current node
ValInfo	Validation Info
Tags	Tags
Destination	Destinationnode
LayerIndex	Index of the Layer

Rückgabe

Boolean value indicating if the validation was successfull

Implementiert NSA.Model.NetworkComponents.ILayer.

6.1.3.6 void NSA.Model.NetworkComponents.Layers.ApplicationLayer.ValidateSend (Workstation *Destination,* Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* int *LayerIndex*)

Validates the layer while sending a packet.

Parameter

Destination	The Destination
CurrentNode	Current Node
ValInfo	Validation Info
Tags	Tags
LayerIndex	The Layer index

Implementiert NSA.Model.NetworkComponents.ILayer.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩Components/Layers/ApplicationLayer.cs

6.2 NSA.Model.NetworkComponents.Connection Klassenreferenz

Class for a connection between two hardwarenodes.

Öffentliche Methoden

- Connection (Hardwarenode Source, Hardwarenode Target)

  Initializes a new instance of the Connection class.
- override bool Equals (object Obj)

  Determines whether the specified System.Object, is equal to this instance.
- bool Equals (Connection Other)

  Equalses the specified other.
- override int GetHashCode ()

  Returns a hash code for this instance.
- int GetPortIndex (Hardwarenode Node)

  Gets the index of the port.

Öffentliche, statische Methoden

- static bool operator== (Connection A, Connection B)

  Implements the operator ==.
- static bool operator!= (Connection A, Connection B)

  Implements the operator !=.

Propertys

- Hardwarenode Start `[get]`

  Gets the start hardwarenode.
- Hardwarenode End `[get]`

  Gets the end hardwarenode.
- string Name `[get]`

  Gets the name (id) of the connection. Every connection has a unique id.

6.2.1 Ausführliche Beschreibung

Class for a connection between two hardwarenodes.

6.2.2 Beschreibung der Konstruktoren und Destruktoren

6.2.2.1 NSA.Model.NetworkComponents.Connection.Connection (Hardwarenode *Source,* Hardwarenode *Target*)

Initializes a new instance of the Connection class.

Parameter

Source	The sourcenode.
Target	The targetnode.

6.2.3 Dokumentation der Elementfunktionen

6.2.3.1 override bool NSA.Model.NetworkComponents.Connection.Equals (object *Obj*)

Determines whether the specified System.Object, is equal to this instance.

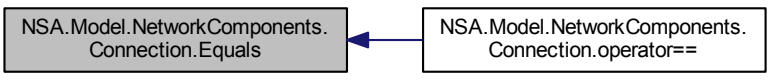
Parameter

Obj	The System.Object to compare with this instance.

Rückgabe

`true` if the specified System.Object is equal to this instance; otherwise, `false`.

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



6.2.3.2 bool NSA.Model.NetworkComponents.Connection.Equals (Connection *Other*)

Equalses the specified other.

Parameter

Other	The other.

Rückgabe

6.2.3.3 override int NSA.Model.NetworkComponents.Connection.GetHashCode ( )

Returns a hash code for this instance.

Rückgabe

A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

6.2.3.4 int NSA.Model.NetworkComponents.Connection.GetPortIndex (Hardwarenode *Node*)

Gets the index of the port.

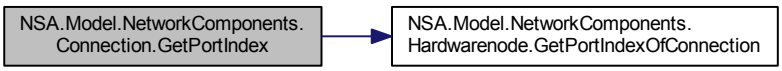
Parameter

Node	The node.

Rückgabe

Portindex

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



6.2.3.5 static bool NSA.Model.NetworkComponents.Connection.operator!= (Connection *A*, Connection *B*) [static]

Implements the operator !=.

Parameter

A	a.
B	The b.

Rückgabe

The result of the operator.

6.2.3.6 static bool NSA.Model.NetworkComponents.Connection.operator== (Connection *A*, Connection *B*) [static]

Implements the operator ==.

Parameter

A	a.
B	The b.

Rückgabe

The result of the operator.

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

6.2.4 Dokumentation der Propertys

6.2.4.1 Hardwarenode NSA.Model.NetworkComponents.Connection.End [get]

Gets the end hardwarenode.

The end.

6.2.4.2 string NSA.Model.NetworkComponents.Connection.Name [get]

Gets the name (id) of the connection. Every connection has a unique id.

The name.

6.2.4.3 Hardwarenode NSA.Model.NetworkComponents.Connection.Start [get]

Gets the start hardwarenode.

The start.

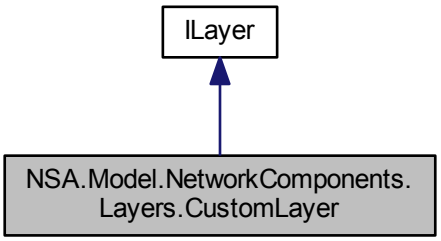
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩ Components/Connection.cs

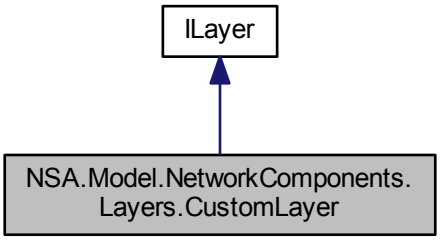
6.3 NSA.Model.NetworkComponents.Layers.CustomLayer Klassenreferenz

Custom-Layer

Klassendiagramm für NSA.Model.NetworkComponents.Layers.CustomLayer:



Zusammengehörigkeiten von NSA.Model.NetworkComponents.Layers.CustomLayer:



Öffentliche Methoden

- CustomLayer (string N, int I)

  Initializes a new instance of the CustomLayer class.
- void ValidateSend (Workstation Destination, Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, int LayerIndex)

  Validates the layer while sending a packet.
- bool ValidateReceive (Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)

  Validates the layer while receiving a packet.
- string GetLayerName ()

  Gets the name of the layer.
- bool SetLayerName (string NewName)

  Sets the name of the layer.
- int GetLayerIndex ()

  Gets the index of the layer.
- void SetLayerIndex (int I)

  Sets the index of the layer.

6.3.1 Ausführliche Beschreibung

Custom-Layer

Siehe auch

NSA.Model.NetworkComponents.ILayer

6.3.2 Beschreibung der Konstruktoren und Destruktoren

6.3.2.1 NSA.Model.NetworkComponents.Layers.CustomLayer.CustomLayer (string *N*, int *I*)

Initializes a new instance of the CustomLayer class.

Parameter

N	The name.
I	The index.

6.3.3 Dokumentation der Elementfunktionen

6.3.3.1 int NSA.Model.NetworkComponents.Layers.CustomLayer.GetLayerIndex ( )

Gets the index of the layer.

Rückgabe

The index

Implementiert NSA.Model.NetworkComponents.ILayer.

6.3.3.2 string NSA.Model.NetworkComponents.Layers.CustomLayer.GetLayerName ( )

Gets the name of the layer.

Rückgabe

The Layername

Implementiert NSA.Model.NetworkComponents.ILayer.

6.3.3.3 void NSA.Model.NetworkComponents.Layers.CustomLayer.SetLayerIndex (int *I*)

Sets the index of the layer.

Parameter

I	The Index.

Implementiert NSA.Model.NetworkComponents.ILayer.

6.3.3.4 bool NSA.Model.NetworkComponents.Layers.CustomLayer.SetLayerName (string *NewName*)

Sets the name of the layer.

Parameter

NewName	The new Name

Rückgabe

Implementiert NSA.Model.NetworkComponents.ILayer.

6.3.3.5 bool NSA.Model.NetworkComponents.Layers.CustomLayer.ValidateReceive (Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* Hardwarenode *Destination,* int *LayerIndex*)

Validates the layer while receiving a packet.

Parameter

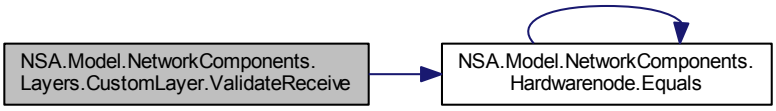
CurrentNode	
ValInfo	
Tags	
Destination	
LayerIndex	

Rückgabe

Boolean value indicating if the validation was successfull

Implementiert NSA.Model.NetworkComponents.ILayer.

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



6.3.3.6 void NSA.Model.NetworkComponents.Layers.CustomLayer.ValidateSend (Workstation *Destination,* Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* int *LayerIndex*)

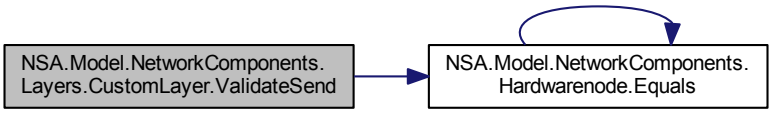
Validates the layer while sending a packet.

Parameter

Destination	The Destination
CurrentNode	Current Node
ValInfo	Validation Info
Tags	Tags
LayerIndex	The Layer index

Implementiert NSA.Model.NetworkComponents.ILayer.

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



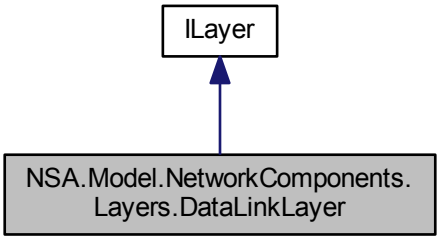
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩ Components/Layers/CustomLayer.cs

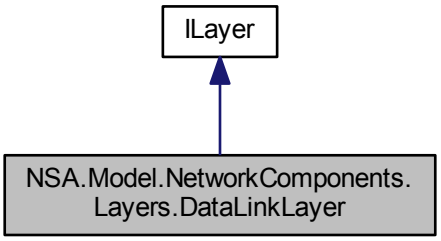
6.4 NSA.Model.NetworkComponents.Layers.DataLinkLayer Klassenreferenz

DataLink-Layer

Klassendiagramm für NSA.Model.NetworkComponents.Layers.DataLinkLayer:



Zusammengehörigkeiten von NSA.Model.NetworkComponents.Layers.DataLinkLayer:



Öffentliche Methoden

- DataLinkLayer (int I)

  Initializes a new instance of the DataLinkLayer class.
- void ValidateSend (Workstation Destination, Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, int LayerIndex)

  Validates the layer while sending a packet.
- bool ValidateReceive (Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)

  Validates the layer while receiving a packet.
- string GetLayerName ()

  Gets the name of the layer.
- bool SetLayerName (string NewName)

  Sets the name of the layer.

- int GetLayerIndex ()

  Gets the index of the layer.
- void SetLayerIndex (int l)

  Sets the index of the layer.

6.4.1 Ausführliche Beschreibung

DataLink-Layer

Siehe auch

NSA.Model.NetworkComponents.ILayer

6.4.2 Beschreibung der Konstruktoren und Destruktoren

6.4.2.1 NSA.Model.NetworkComponents.Layers.DataLinkLayer.DataLinkLayer (int *l*)

Initializes a new instance of the DataLinkLayer class.

Parameter

l	The index.

6.4.3 Dokumentation der Elementfunktionen

6.4.3.1 int NSA.Model.NetworkComponents.Layers.DataLinkLayer.GetLayerIndex ( )

Gets the index of the layer.

Rückgabe

The index

Implementiert NSA.Model.NetworkComponents.ILayer.

6.4.3.2 string NSA.Model.NetworkComponents.Layers.DataLinkLayer.GetLayerName ( )

Gets the name of the layer.

Rückgabe

The Layername

Implementiert NSA.Model.NetworkComponents.ILayer.

6.4.3.3 void NSA.Model.NetworkComponents.Layers.DataLinkLayer.SetLayerIndex (int *l*)

Sets the index of the layer.

Parameter

I	The Index.

Implementiert NSA.Model.NetworkComponents.ILayer.

6.4.3.4 bool NSA.Model.NetworkComponents.Layers.DataLinkLayer.SetLayerName (string *NewName*)

Sets the name of the layer.

Parameter

NewName	New Name

Rückgabe

Implementiert NSA.Model.NetworkComponents.ILayer.

6.4.3.5 bool NSA.Model.NetworkComponents.Layers.DataLinkLayer.ValidateReceive (Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* Hardwarenode *Destination,* int *LayerIndex*)

Validates the layer while receiving a packet.

Parameter

CurrentNode	Current node
ValInfo	Validation Info
Tags	Tags
Destination	Destinationnode
LayerIndex	Index of the Layer

Rückgabe

Boolean value indicating if the validation was successfull

Implementiert NSA.Model.NetworkComponents.ILayer.

6.4.3.6 void NSA.Model.NetworkComponents.Layers.DataLinkLayer.ValidateSend (Workstation *Destination,* Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* int *LayerIndex*)

Validates the layer while sending a packet.

Parameter

Destination	The Destination

Parameter

CurrentNode	Current Node
ValInfo	Validation Info
Tags	Tags
LayerIndex	The Layer index

Implementiert NSA.Model.NetworkComponents.ILayer.

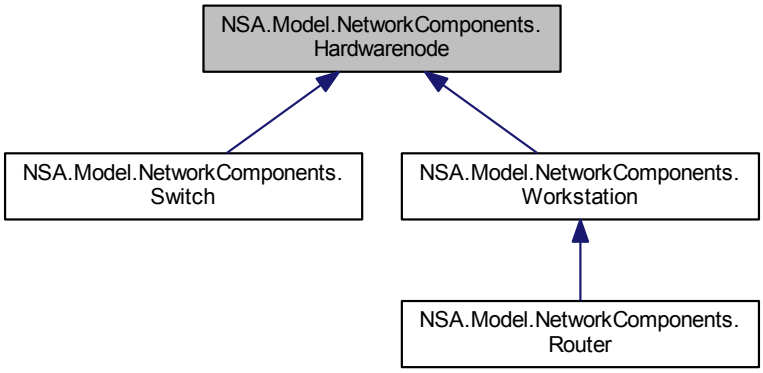
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩ Components/Layers/DataLinkLayer.cs

6.5 NSA.Model.NetworkComponents.Hardwarenode Klassenreferenz

Implements the basis class for hardwarenodes.

Klassendiagramm für NSA.Model.NetworkComponents.Hardwarenode:



Öffentliche Methoden

- Hardwarenode (string N)

  Initializes a new instance of the Hardwarenode class.
- virtual Interface AddInterface (IPAddress Ip, IPAddress Subnetmask, int PortNum=-1)

  Adds a new interface with the given IP and subnetmask
- void RemoveInterface (string InterfaceName)

  Removes the interface with the given name.
- int GetInterfaceCount ()

  Gets the interface count.
- virtual void SetInterface (string Ifacename, IPAddress Ip, IPAddress Mask)

  Sets the interface.

- virtual bool HasInterface (string IfaceName)

  Determines if there is an Interface with the specified name.
- void AddConnection (string IfaceName, Connection Con)

  Adds a connection.
- virtual void RemoveConnection (string IfaceName)

  Removes a connection.
- Connection GetConnectionAtPort (string IfaceName)

  Gets the connection at port.
- bool InterfaceIsUsed (string InterfaceName)

  Interfaces the is used.
- virtual bool HasIp (IPAddress Ip)

  Checks if the Hardwarenode has the IP
- virtual List< Hardwarenode > Send (Hardwarenode Destination, Dictionary< string, object > Tags, ValidationInfo ValInfo)

  Hardwarenode sends the package to specified destination.
- virtual bool Receive (Dictionary< string, object > Tags, ValidationInfo ValInfo, Hardwarenode Destination)

  Hardwarenode receives the package.
- override bool Equals (object Obj)

  Determines whether the specified System.Object, is equal to this instance.
- bool Equals (Hardwarenode Other)

  Equalses the specified other.
- override int GetHashCode ()

  Returns a hash code for this instance.
- int GetPortIndexOfConnection (Connection C)

  Gets the port index of connection.

Öffentliche, statische Methoden

- static bool operator== (Hardwarenode A, Hardwarenode B)

  Implements the operator ==.
- static bool operator!= (Hardwarenode A, Hardwarenode B)

  Implements the operator !=.

Geschützte Methoden

- int getNewInterfaceNumber ()

  Gets the new interface number.

Propertys

- Layerstack Layerstack `[get]`

  Gets the layerstack.
- Dictionary< string, Connection > Connections = new Layerstack() `[get, protected set]`

  Gets or sets the connections.
- string Name = new Dictionary<string, Connection>() `[get, set]`

  Gets or sets the name of the hardwarenode.
- List< Interface > Interfaces `[get, protected set]`

  Gets or sets the interfaces.

6.5.1 Ausführliche Beschreibung

Implements the basis class for hardwarenodes.

6.5.2 Beschreibung der Konstruktoren und Destruktoren

6.5.2.1 NSA.Model.NetworkComponents.Hardwarenode.Hardwarenode (string *N*)

Initializes a new instance of the Hardwarenode class.

Parameter

N	The n.

6.5.3 Dokumentation der Elementfunktionen

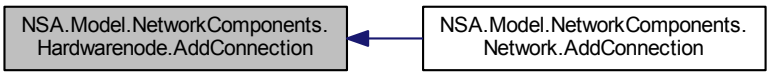
6.5.3.1 void NSA.Model.NetworkComponents.Hardwarenode.AddConnection (string *IfaceName,* Connection *Con*)

Adds a connection.

Parameter

IfaceName	Name of the interface where the connection should be added.
Con	The connection to be added.

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



6.5.3.2 virtual Interface NSA.Model.NetworkComponents.Hardwarenode.AddInterface (IPAddress *Ip,* IPAddress *Subnetmask,* int *PortNum* = −1) [virtual]

Adds a new interface with the given IP and subnetmask

Parameter

Ip	The IP of the interface. Ignored if used with switch
Subnetmask	The subnetmask. Ignored if used with switch
PortNum	Number of port. Only for project loading purpose.

Rückgabe

The newly added Interface

Erneute Implementation in NSA.Model.NetworkComponents.Switch.

6.5.3.3 override bool NSA.Model.NetworkComponents.Hardwarenode.Equals (object *Obj*)

Determines whether the specified System.Object, is equal to this instance.

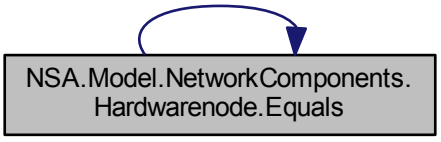
Parameter

Obj	The System.Object to compare with this instance.

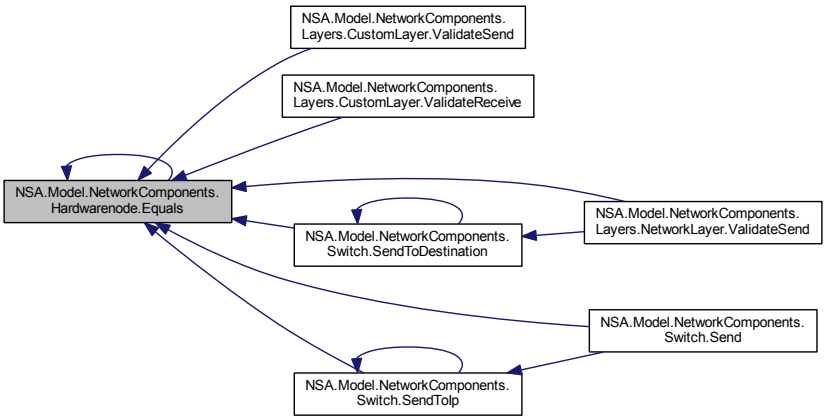
Rückgabe

`true` if the specified System.Object is equal to this instance; otherwise, `false`.

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



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



6.5.3.4 bool NSA.Model.NetworkComponents.Hardwarenode.Equals (Hardwarenode *Other*)

Equalses the specified other.

Parameter

Other	The other.

Rückgabe

6.5.3.5 Connection NSA.Model.NetworkComponents.Hardwarenode.GetConnectionAtPort (string *IfaceName*)

Gets the connection at port.

Parameter

IfaceName	Name of the port.

Rückgabe

6.5.3.6 override int NSA.Model.NetworkComponents.Hardwarenode.GetHashCode ( )

Returns a hash code for this instance.

Rückgabe

A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.

6.5.3.7 int NSA.Model.NetworkComponents.Hardwarenode.GetInterfaceCount ( )

Gets the interface count.

Rückgabe

int: interface count

6.5.3.8 int NSA.Model.NetworkComponents.Hardwarenode.getNewInterfaceNumber ( ) [protected]

Gets the new interface number.

Rückgabe

int: number for next interface

6.5.3.9 int NSA.Model.NetworkComponents.Hardwarenode.GetPortIndexOfConnection (Connection *C*)

Gets the port index of connection.

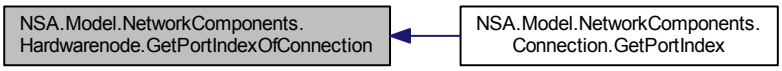
Parameter

C	The connection.

Rückgabe

Portindex

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



6.5.3.10 virtual bool NSA.Model.NetworkComponents.Hardwarenode.HasInterface (string *IfaceName*) [virtual]

Determines if there is an Interface with the specified name.

Parameter

IfaceName	Name of the iface.

Rückgabe

6.5.3.11 virtual bool NSA.Model.NetworkComponents.Hardwarenode.HasIp (IPAddress *Ip*) [virtual]

Checks if the Hardwarenode has the IP

Parameter

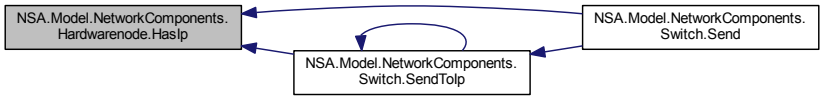
Ip	The ip.

Rückgabe

A bool

Erneute Implementation in NSA.Model.NetworkComponents.Workstation.

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



6.5.3.12 bool NSA.Model.NetworkComponents.Hardwarenode.InterfaceIsUsed (string *InterfaceName*)

Interfaces the is used.

Parameter

InterfaceName	Name of the interface.

Rückgabe

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

6.5.3.13 static bool NSA.Model.NetworkComponents.Hardwarenode.operator!= (Hardwarenode *A*, Hardwarenode *B*) [static]

Implements the operator !=.

Parameter

A	a.
B	The b.

Rückgabe

The result of the operator.

6.5.3.14 static bool NSA.Model.NetworkComponents.Hardwarenode.operator== (Hardwarenode *A*, Hardwarenode *B*) [static]

Implements the operator ==.

Parameter

A	a.
B	The b.

Rückgabe

The result of the operator.

6.5.3.15 virtual bool NSA.Model.NetworkComponents.Hardwarenode.Receive (Dictionary< string, object > *Tags*, ValidationInfo *ValInfo*, Hardwarenode *Destination*) [virtual]

Hardwarenode receives the package.

Parameter

Tags	Optional tags.
ValInfo	The validation Info
Destination	The destination.

Rückgabe

If the Hardwarenode could receive the package

Erneute Implementation in NSA.Model.NetworkComponents.Workstation.

6.5.3.16 virtual void NSA.Model.NetworkComponents.Hardwarenode.RemoveConnection (string *IfaceName*) [virtual]

Removes a connection.

Parameter

IfaceName	Name of the interface where the connection should be removed.

6.5.3.17 void NSA.Model.NetworkComponents.Hardwarenode.RemoveInterface (string *InterfaceName*)

Removes the interface with the given name.

Parameter

InterfaceName	The Interfacename.

6.5.3.18 virtual List<Hardwarenode> NSA.Model.NetworkComponents.Hardwarenode.Send (Hardwarenode *Destination,* Dictionary< string, object > *Tags,* ValidationInfo *ValInfo*) [virtual]

Hardwarenode sends the package to specified destination.

Parameter

Destination	The destination.
Tags	Optional tags.
ValInfo	

Rückgabe

The Hardwarenode which received the package or null if an error occured

Erneute Implementation in NSA.Model.NetworkComponents.Workstation und NSA.Model.NetworkComponents.↩ Switch.

6.5.3.19 virtual void NSA.Model.NetworkComponents.Hardwarenode.SetInterface (string *Ifacename,* IPAddress *Ip,* IPAddress *Mask*) [virtual]

Sets the interface.

Parameter

Ifacename	The name of the Interface.
Ip	The new ip.
Mask	The new subnetmask.

Rückgabe

bool: false if the interface could not be found, otherwise true

6.5.4 Dokumentation der Propertys

6.5.4.1 Dictionary<string, Connection> NSA.Model.NetworkComponents.Hardwarenode.Connections = new Layerstack() [get], [protected set]

Gets or sets the connections.

The connections.

6.5.4.2 List<Interface> NSA.Model.NetworkComponents.Hardwarenode.Interfaces [get], [protected set]

Gets or sets the interfaces.

The interfaces.

6.5.4.3 Layerstack NSA.Model.NetworkComponents.Hardwarenode.Layerstack [get]

Gets the layerstack.

The layerstack.

6.5.4.4 string NSA.Model.NetworkComponents.Hardwarenode.Name = new Dictionary<string, Connection>() [get], [set]

Gets or sets the name of the hardwarenode.

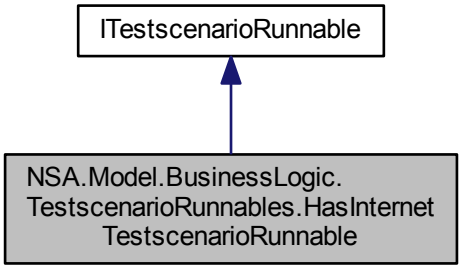
The name.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

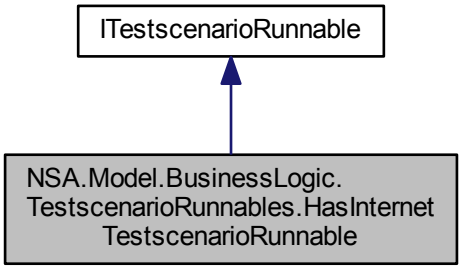
- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩ Components/Hardwarenode.cs

6.6 NSA.Model.BusinessLogic.TestscenarioRunnables.HasInternetTestscenarioRunnable Klassenreferenz

Klassendiagramm für NSA.Model.BusinessLogic.TestscenarioRunnables.HasInternetTestscenarioRunnable:



Zusammengehörigkeiten von NSA.Model.BusinessLogic.TestscenarioRunnables.HasInternetTestscenario↩ Runnable:



Öffentliche Methoden

- HasInternetTestscenarioRunnable (Rule Rule)

  Initializes a new instance of the NSA.Model.BusinessLogic.TestscenarioRunnables.HasInternetTestscenarioRunnable class.
- List< Simulation > Run ()

  runs all simulations for a given rule

6.6.1 Beschreibung der Konstruktoren und Destruktoren

6.6.1.1 NSA.Model.BusinessLogic.TestscenarioRunnables.HasInternetTestscenarioRunnable.HasInternetTestscenarioRunnable (Rule *Rule*)

Initializes a new instance of the NSA.Model.BusinessLogic.TestscenarioRunnables.HasInternetTestscenarioRunnable class.

Parameter

Rule	Rule object

6.6.2 Dokumentation der Elementfunktionen

6.6.2.1 List<Simulation> NSA.Model.BusinessLogic.TestscenarioRunnables.HasInternetTestscenarioRunnable.Run ( )

runs all simulations for a given rule

Rückgabe

simulations that failed

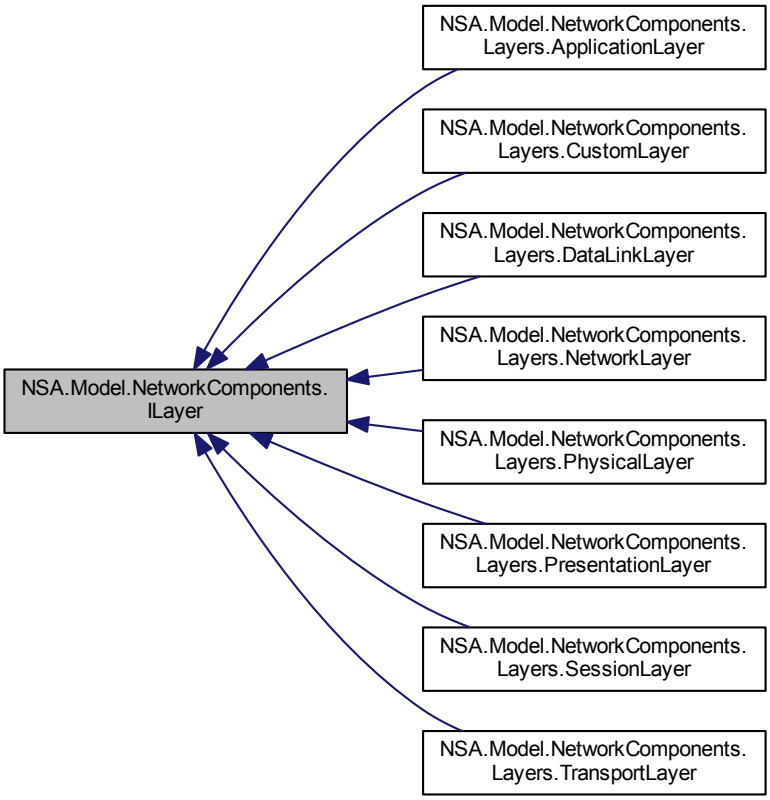
Implementiert NSA.Model.BusinessLogic.TestscenarioRunnables.ITestscenarioRunnable.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/BusinessLogic/TestscenarioRunnables/HasInternetTestscenarioRunnable.cs

6.7 NSA.Model.NetworkComponents.ILayer Schnittstellenreferenz

Klassendiagramm für NSA.Model.NetworkComponents.ILayer:



Öffentliche Methoden

- void ValidateSend (Workstation Destination, Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, int LayerIndex)

  Validates the layer while sending a packet.
- bool ValidateReceive (Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)

  Validates the layer while receiving a packet.
- string GetLayerName ()

  Gets the name of the layer.
- bool SetLayerName (string NewName)

  Sets the name of the layer.
- int GetLayerIndex ()

  Gets the index of the layer.
- void SetLayerIndex (int I)

  Sets the index of the layer.

6.7.1 Dokumentation der Elementfunktionen

6.7.1.1 int NSA.Model.NetworkComponents.ILayer.GetLayerIndex ( )

Gets the index of the layer.

Rückgabe

Implementiert in NSA.Model.NetworkComponents.Layers.DataLinkLayer, NSA.Model.NetworkComponents.Layers.CustomLayer, NSA.Model.NetworkComponents.Layers.NetworkLayer, NSA.Model.NetworkComponents.Layers.ApplicationLayer, NSA.Model.NetworkComponents.Layers.PhysicalLayer, NSA.Model.NetworkComponents.Layers.PresentationLayer, NSA.Model.NetworkComponents.Layers.SessionLayer und NSA.Model.NetworkComponents.Layers.TransportLayer.

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

6.7.1.2 string NSA.Model.NetworkComponents.ILayer.GetLayerName ( )

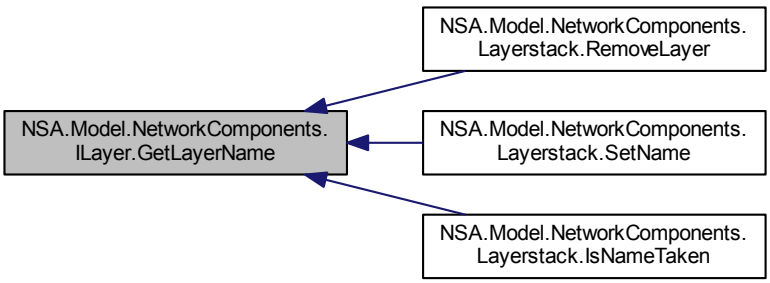
Gets the name of the layer.

Rückgabe

The Layername

Implementiert in NSA.Model.NetworkComponents.Layers.DataLinkLayer, NSA.Model.NetworkComponents.Layers.CustomLayer, NSA.Model.NetworkComponents.Layers.NetworkLayer, NSA.Model.NetworkComponents.Layers.ApplicationLayer, NSA.Model.NetworkComponents.Layers.PhysicalLayer, NSA.Model.NetworkComponents.Layers.PresentationLayer, NSA.Model.NetworkComponents.Layers.SessionLayer und NSA.Model.NetworkComponents.Layers.TransportLayer.

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



6.7.1.3 void NSA.Model.NetworkComponents.ILayer.SetLayerIndex (int *I*)

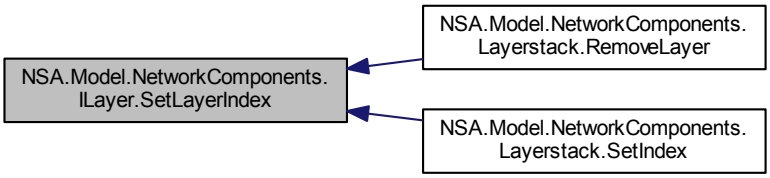
Sets the index of the layer.

Parameter

I	The Index.

Implementiert in NSA.Model.NetworkComponents.Layers.DataLinkLayer, NSA.Model.NetworkComponents.Layers.CustomLayer, NSA.Model.NetworkComponents.Layers.NetworkLayer, NSA.Model.NetworkComponents.Layers.ApplicationLayer, NSA.Model.NetworkComponents.Layers.PhysicalLayer, NSA.Model.NetworkComponents.Layers.PresentationLayer, NSA.Model.NetworkComponents.Layers.SessionLayer und NSA.Model.NetworkComponents.Layers.TransportLayer.

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



6.7.1.4 bool NSA.Model.NetworkComponents.ILayer.SetLayerName (string *NewName*)

Sets the name of the layer.

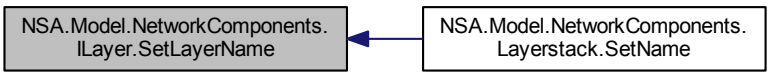
Parameter

NewName	

Rückgabe

Implementiert in NSA.Model.NetworkComponents.Layers.DataLinkLayer, NSA.Model.NetworkComponents.Layers.CustomLayer, NSA.Model.NetworkComponents.Layers.NetworkLayer, NSA.Model.NetworkComponents.Layers.ApplicationLayer, NSA.Model.NetworkComponents.Layers.PhysicalLayer, NSA.Model.NetworkComponents.Layers.PresentationLayer, NSA.Model.NetworkComponents.Layers.SessionLayer und NSA.Model.NetworkComponents.Layers.TransportLayer.

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



6.7.1.5 bool NSA.Model.NetworkComponents.ILayer.ValidateReceive (Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* Hardwarenode *Destination,* int *LayerIndex*)

Validates the layer while receiving a packet.

Parameter

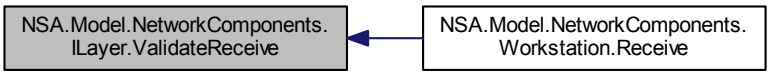
CurrentNode	
ValInfo	
Tags	
Destination	
LayerIndex	

Rückgabe

Boolean value indicating if the validation was successfull

Implementiert in NSA.Model.NetworkComponents.Layers.DataLinkLayer, NSA.Model.NetworkComponents.Layers.CustomLayer, NSA.Model.NetworkComponents.Layers.NetworkLayer, NSA.Model.NetworkComponents.Layers.ApplicationLayer, NSA.Model.NetworkComponents.Layers.PhysicalLayer, NSA.Model.NetworkComponents.Layers.PresentationLayer, NSA.Model.NetworkComponents.Layers.SessionLayer und NSA.Model.NetworkComponents.Layers.TransportLayer.

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



6.7.1.6 void NSA.Model.NetworkComponents.ILayer.ValidateSend (Workstation *Destination,* Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* int *LayerIndex*)

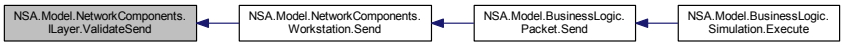
Validates the layer while sending a packet.

Parameter

Destination	
CurrentNode	
ValInfo	
Tags	
LayerIndex	

Implementiert in NSA.Model.NetworkComponents.Layers.NetworkLayer, NSA.Model.NetworkComponents.↩ Layers.ApplicationLayer, NSA.Model.NetworkComponents.Layers.PhysicalLayer, NSA.Model.NetworkComponents.↩ Layers.PresentationLayer, NSA.Model.NetworkComponents.Layers.SessionLayer, NSA.Model.NetworkComponents.↩ Layers.TransportLayer, NSA.Model.NetworkComponents.Layers.CustomLayer und NSA.Model.Network↩ Components.Layers.DataLinkLayer.

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



Die Dokumentation für diese Schnittstelle wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩ Components/ILayer.cs

6.8 NSA.Model.NetworkComponents.Interface Klassenreferenz

Class for a interface of an hardwarenode.

Öffentliche Methoden

- Interface (IPAddress Ip, IPAddress Mask, int Number)

  Initializes a new instance of the Interface class.
- void SetInterface (IPAddress Ip, IPAddress Mask)

  Sets the interface.

Öffentliche Attribute

- const string NamePrefix = "eth"

  The name prefix of the interfacename.

Propertys

- string Name `[get]`

  Gets the name of the interface
- IPAddress IpAddress `[get, set]`

  Gets or sets the ip address.
- IPAddress Subnetmask `[get, set]`

  Gets or sets the subnetmask.

6.8.1 Ausführliche Beschreibung

Class for a interface of an hardwarenode.

6.8.2 Beschreibung der Konstruktoren und Destruktoren

6.8.2.1 NSA.Model.NetworkComponents.Interface.Interface (IPAddress *Ip,* IPAddress *Mask,* int *Number*)

Initializes a new instance of the Interface class.

Parameter

Ip	The ip address of the interface.
Mask	The corresponding subnetmask.
Number	The number (e.g. 0 for eth0).

6.8.3 Dokumentation der Elementfunktionen

6.8.3.1 void NSA.Model.NetworkComponents.Interface.SetInterface (IPAddress *Ip,* IPAddress *Mask*)

Sets the interface.

Parameter

Ip	The new ip.
Mask	The new subnetmask.

6.8.4 Dokumentation der Datenelemente

6.8.4.1 const string NSA.Model.NetworkComponents.Interface.NamePrefix = "eth"

The name prefix of the interfacename.

6.8.5 Dokumentation der Propertys

6.8.5.1 IPAddress NSA.Model.NetworkComponents.Interface.IpAddress [get],[set]

Gets or sets the ip address.

The ip address.

6.8.5.2 string NSA.Model.NetworkComponents.Interface.Name [get]

Gets the name of the interface

The name.

6.8.5.3 IPAddress NSA.Model.NetworkComponents.Interface.Subnetmask [get], [set]

Gets or sets the subnetmask.

The subnetmask.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Interface.cs

6.9 NSA.Model.NetworkComponents.Helper_Classes.IPAddressExtensions Klassenreferenz

Öffentliche, statische Methoden

- static IPAddress GetBroadcastAddress (this IPAddress address, IPAddress subnetMask)
- static IPAddress GetNetworkAddress (this IPAddress address, IPAddress subnetMask)
- static bool IsInSameSubnet (this IPAddress address2, IPAddress address, IPAddress subnetMask)
- static bool IsValidSubnetMask (this IPAddress Subnetmask)

6.9.1 Dokumentation der Elementfunktionen

6.9.1.1 static IPAddress NSA.Model.NetworkComponents.Helper_Classes.IPAddressExtensions.GetBroadcastAddress (this IPAddress *address,* IPAddress *subnetMask*) [static]

6.9.1.2 static IPAddress NSA.Model.NetworkComponents.Helper_Classes.IPAddressExtensions.GetNetworkAddress (this IPAddress *address,* IPAddress *subnetMask*) [static]

6.9.1.3 static bool NSA.Model.NetworkComponents.Helper_Classes.IPAddressExtensions.IsInSameSubnet (this IPAddress *address2,* IPAddress *address,* IPAddress *subnetMask*) [static]

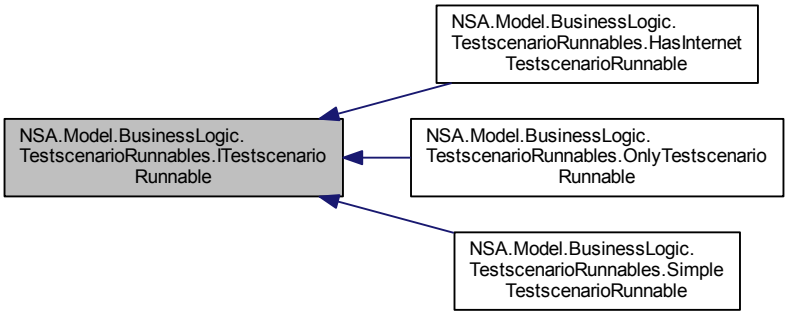
6.9.1.4 static bool NSA.Model.NetworkComponents.Helper_Classes.IPAddressExtensions.IsValidSubnetMask (this IPAddress *Subnetmask*) [static]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Helper Classes/IPAddressExtensions.cs

6.10 NSA.Model.BusinessLogic.TestscenarioRunnables.ITestscenarioRunnable Schnittstellenreferenz

Klassendiagramm für NSA.Model.BusinessLogic.TestscenarioRunnables.ITestscenarioRunnable:



Öffentliche Methoden

- List< Simulation > Run ()

  runs all simulations for a given rule

6.10.1 Dokumentation der Elementfunktionen

6.10.1.1 List<Simulation> NSA.Model.BusinessLogic.TestscenarioRunnables.ITestscenarioRunnable.Run ( )

runs all simulations for a given rule

Rückgabe

simulations that failed

Implementiert in NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable, NSA.Model.↩ BusinessLogic.TestscenarioRunnables.HasInternetTestscenarioRunnable und NSA.Model.BusinessLogic.↩ TestscenarioRunnables.SimpleTestscenarioRunnable.

Die Dokumentation für diese Schnittstelle wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↩ Logic/TestscenarioRunnables/ITestscenarioRunnable.cs

6.11 NSA.Model.NetworkComponents.Layerstack Klassenreferenz

Layerstack

Öffentliche Methoden

- Layerstack ()

  Initializes a new instance of the Layerstack class.
- void AddLayer (ILayer Lay)

  Adds a layer to the stack.
- void RemoveLayer (string Name)

  Removes a layer from the stack.
- int GetSize ()

  Returns the size of the layerstack.
- ILayer GetLayer (int Index)

  Returns the layer at the index.
- void InsertAt (int Index, ILayer Layer)

  Inserts a layer at the index.
- void SetIndex (string Name, int NewIndex)

  Sets the index.
- ILayer GetLayerByName (string Name)

  Gets the layer with the name.
- bool SetName (string OldName, string NewName)

  Sets the name.
- List< ILayer > GetAllLayers ()

  Gets all layers.
- bool IsNameTaken (string Name)

  Determines whether the name is taken or not.
- string CreateUniqueName ()

  Creates a new unique name.

6.11.1 Ausführliche Beschreibung

Layerstack

6.11.2 Beschreibung der Konstruktoren und Destruktoren

6.11.2.1 NSA.Model.NetworkComponents.Layerstack.Layerstack ( )

Initializes a new instance of the Layerstack class.

6.11.3 Dokumentation der Elementfunktionen

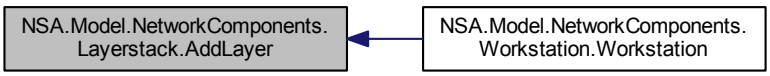
6.11.3.1 void NSA.Model.NetworkComponents.Layerstack.AddLayer (ILayer *Lay*)

Adds a layer to the stack.

Parameter

Lay	The layer to be added.

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



6.11.3.2 string NSA.Model.NetworkComponents.Layerstack.CreateUniqueName ( )

Creates a new unique name.

Rückgabe

6.11.3.3 List<ILayer> NSA.Model.NetworkComponents.Layerstack.GetAllLayers ( )

Gets all layers.

Rückgabe

6.11.3.4 ILayer NSA.Model.NetworkComponents.Layerstack.GetLayer (int *Index*)

Returns the layer at the index.

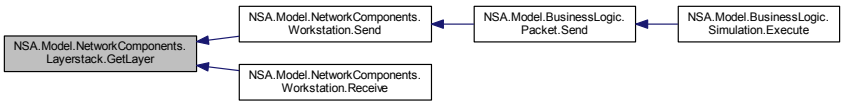
Parameter

Index	The index.

Rückgabe

The layer

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



6.11.3.5 ILayer NSA.Model.NetworkComponents.Layerstack.GetLayerByName (string *Name*)

Gets the layer with the name.

Parameter

Name	The name.

Rückgabe

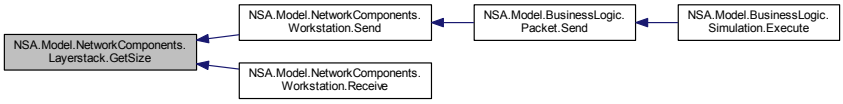
6.11.3.6 int NSA.Model.NetworkComponents.Layerstack.GetSize ( )

Returns the size of the layerstack.

Rückgabe

The size

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



6.11.3.7 void NSA.Model.NetworkComponents.Layerstack.InsertAt (int *Index,* ILayer *Layer*)

Inserts a layer at the index.

Parameter

Index	The index.
Layer	The layer.

6.11.3.8 bool NSA.Model.NetworkComponents.Layerstack.IsNameTaken (string *Name*)

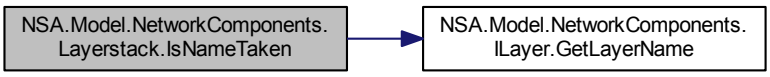
Determines whether the name is taken or not.

Parameter

Name	The name.

Rückgabe

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



6.11.3.9 void NSA.Model.NetworkComponents.Layerstack.RemoveLayer (string *Name*)

Removes a layer from the stack.

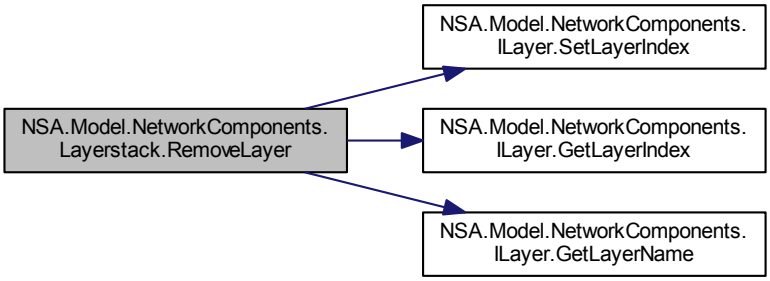
Parameter

Name	The name.

Ausnahmebehandlung

System.InvalidOperationException	Layer with the name: + name + does not exist.

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



6.11.3.10 void NSA.Model.NetworkComponents.Layerstack.SetIndex (string *Name,* int *NewIndex*)

Sets the index.

Parameter

Name	The name.
NewIndex	The new index.

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

6.11.3.11 bool NSA.Model.NetworkComponents.Layerstack.SetName (string *OldName,* string *NewName*)

Sets the name.

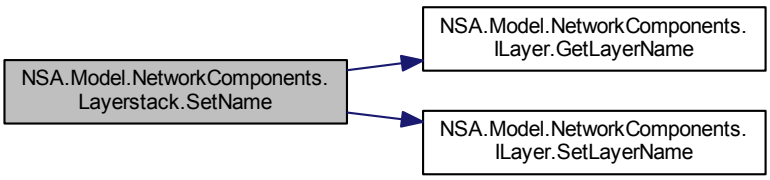
Parameter

OldName	The old name.
NewName	The new name.

Rückgabe

True if it worked. False if the newName is already taken by another layer or when there is no layer with the old name

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



Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩ Components/Layerstack.cs

6.12 NSA.Model.NetworkComponents.Network Klassenreferenz

Öffentliche Methoden

- Network ()
- Hardwarenode GetHardwarenodeByName (string Name)

  Returns the Hardwarenode with the name.
- void AddHardwarenode (Hardwarenode NewNode)

  Adds a hardwarenode.
- bool AddConnection (string StartNodeInterfaceName, string EndNodeInterfaceName, Connection NewConnection)

  Adds the connection.
- void RemoveHardwarnode (string Name)

  Removes the hardwarnode.
- void RemoveConnection (string ConnectionName)

  Removes the connection.
- Hardwarenode GetWorkstationByIp (IPAddress Ip)

  Gets the workstation by ip.
- List< Hardwarenode > GetAllHardwarenodes ()

  Gets all hardwarenodes.
- List< Workstation > GetAllWorkstations ()

  Gets all workstations.
- Connection GetConnectionByName (string Name)

  Gets the name of the connection by.
- List< Router > GetRouters ()

  Gets the routers with internetconnection.
- List< Connection > GetAllConnections ()
- List< Workstation > GetHardwareNodesForSubnet (string Subnetmask)

  Gets all hardwarenodes belonging to a subnet.

Propertys

- List< Connection > Connections `[get]`

6.12.1 Beschreibung der Konstruktoren und Destruktoren

6.12.1.1 NSA.Model.NetworkComponents.Network.Network ( )

6.12.2 Dokumentation der Elementfunktionen

6.12.2.1 bool NSA.Model.NetworkComponents.Network.AddConnection (string *StartNodeInterfaceName,* string *EndNodeInterfaceName,* Connection *NewConnection*)

Adds the connection.

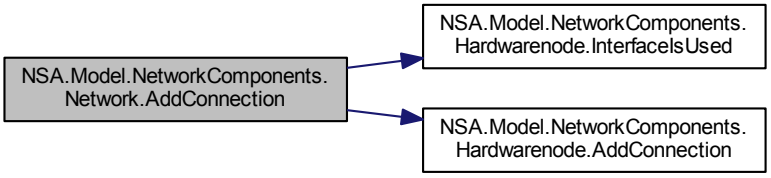
Parameter

StartNodeInterfaceName	Start name of the node interface.
EndNodeInterfaceName	End name of the node interface.
NewConnection	The new connection.

Rückgabe

True on success, false if the connection could not be added because the connection already exists or the connection contains an invalid start- or end-node or if the interface of the start- or endnode is already used.

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



6.12.2.2 void NSA.Model.NetworkComponents.Network.AddHardwarenode (Hardwarenode *NewNode*)

Adds a hardwarenode.

Parameter

NewNode	The new node.

6.12.2.3 List<Connection> NSA.Model.NetworkComponents.Network.GetAllConnections ( )

6.12.2.4 List<Hardwarenode> NSA.Model.NetworkComponents.Network.GetAllHardwarenodes ( )

Gets all hardwarenodes.

Rückgabe

all Hardwarenodes

6.12.2.5 List<Workstation> NSA.Model.NetworkComponents.Network.GetAllWorkstations ( )

Gets all workstations.

Rückgabe

all Workstations

6.12.2.6 Connection NSA.Model.NetworkComponents.Network.GetConnectionByName (string *Name*)

Gets the name of the connection by.

Parameter

Name	The name.

Rückgabe

the connection with its name

6.12.2.7 Hardwarenode NSA.Model.NetworkComponents.Network.GetHardwarenodeByName (string *Name*)

Returns the Hardwarenode with the name.

Parameter

Name	The name.

Rückgabe

The Hardwarenode with this name or default value

6.12.2.8 List<Workstation> NSA.Model.NetworkComponents.Network.GetHardwareNodesForSubnet (string *Subnetmask*)

Gets all hardwarenodes belonging to a subnet.

Parameter

Subnetmask	The subnetmask.

Rückgabe

A list of hardwarenodes who belong to the subnet.

6.12.2.9 List<Router> NSA.Model.NetworkComponents.Network.GetRouters ( )

Gets the routers with internetconnection.

Rückgabe

A List of routers

6.12.2.10 Hardwarenode NSA.Model.NetworkComponents.Network.GetWorkstationByIp (IPAddress *Ip*)

Gets the workstation by ip.

Parameter

Ip	The ip.

Rückgabe

6.12.2.11 void NSA.Model.NetworkComponents.Network.RemoveConnection (string *ConnectionName*)

Removes the connection.

Parameter

ConnectionName	Name of the connection.

6.12.2.12 void NSA.Model.NetworkComponents.Network.RemoveHardwarnode (string *Name*)

Removes the hardwarnode.

Parameter

Name	The name.

6.12.3 Dokumentation der Propertys

6.12.3.1 List<Connection> NSA.Model.NetworkComponents.Network.Connections [get]

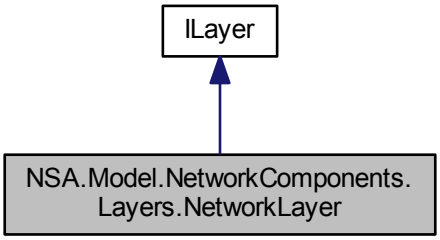
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩ Components/Network.cs

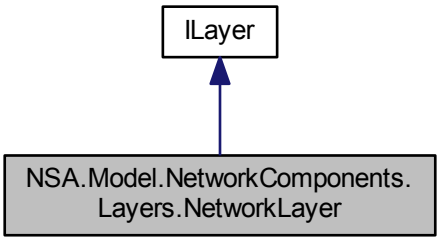
6.13 NSA.Model.NetworkComponents.Layers.NetworkLayer Klassenreferenz

Network-Layer

Klassendiagramm für NSA.Model.NetworkComponents.Layers.NetworkLayer:



Zusammengehörigkeiten von NSA.Model.NetworkComponents.Layers.NetworkLayer:



Öffentliche Methoden

- NetworkLayer (int I)

  Initializes a new instance of the NetworkLayer class.
- bool ValidateReceive (Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)

  Validates the layer while receiving a packet.
- string GetLayerName ()

  Gets the name of the layer.
- bool SetLayerName (string NewName)

  Sets the name of the layer.
- int GetLayerIndex ()

  Gets the index of the layer.
- void SetLayerIndex (int I)

  Sets the index of the layer.
- void ValidateSend (Workstation Destination, Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, int LayerIndex)

  Validates the layer while sending a packet.

6.13.1 Ausführliche Beschreibung

Network-Layer

Siehe auch

NSA.Model.NetworkComponents.ILayer

6.13.2 Beschreibung der Konstruktoren und Destruktoren

6.13.2.1 NSA.Model.NetworkComponents.Layers.NetworkLayer.NetworkLayer (int *I*)

Initializes a new instance of the NetworkLayer class.

Parameter

I	The index.

6.13.3 Dokumentation der Elementfunktionen

6.13.3.1 int NSA.Model.NetworkComponents.Layers.NetworkLayer.GetLayerIndex ( )

Gets the index of the layer.

Rückgabe

The index

Implementiert NSA.Model.NetworkComponents.ILayer.

6.13.3.2 string NSA.Model.NetworkComponents.Layers.NetworkLayer.GetLayerName ( )

Gets the name of the layer.

Rückgabe

The Layername

Implementiert NSA.Model.NetworkComponents.ILayer.

6.13.3.3 void NSA.Model.NetworkComponents.Layers.NetworkLayer.SetLayerIndex (int *I*)

Sets the index of the layer.

Parameter

I	The Index.

Implementiert NSA.Model.NetworkComponents.ILayer.

6.13.3.4 bool NSA.Model.NetworkComponents.Layers.NetworkLayer.SetLayerName (string *NewName*)

Sets the name of the layer.

Parameter

NewName	New Name

Rückgabe

Implementiert NSA.Model.NetworkComponents.ILayer.

6.13.3.5 bool NSA.Model.NetworkComponents.Layers.NetworkLayer.ValidateReceive (Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* Hardwarenode *Destination,* int *LayerIndex*)

Validates the layer while receiving a packet.

Parameter

CurrentNode	Current node
ValInfo	Validation Info
Tags	Tags
Destination	Destinationnode
LayerIndex	Index of the Layer

Rückgabe

Boolean value indicating if the validation was successfull

Implementiert NSA.Model.NetworkComponents.ILayer.

6.13.3.6 void NSA.Model.NetworkComponents.Layers.NetworkLayer.ValidateSend (Workstation *Destination,* Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* int *LayerIndex*)

Validates the layer while sending a packet.

Parameter

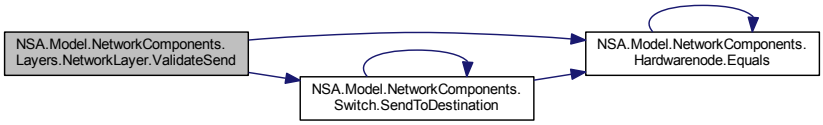
Destination	The Destination

Parameter

CurrentNode	Current Node
ValInfo	Validation Info
Tags	Tags
LayerIndex	The Layer index

Implementiert NSA.Model.NetworkComponents.ILayer.

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

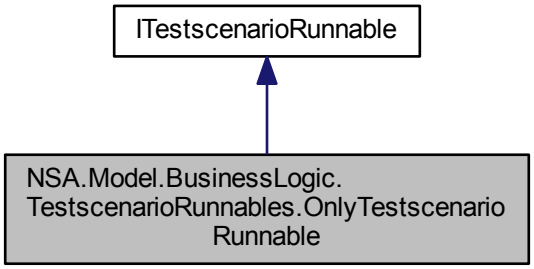


Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

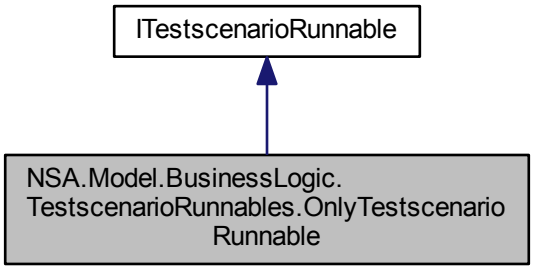
- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩ Components/Layers/NetworkLayer.cs

6.14 NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable Klassenreferenz

Klassendiagramm für NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable:



Zusammengehörigkeiten von NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable:



Öffentliche Methoden

- OnlyTestscenarioRunnable (Rule Rule, Network N)

  Initializes a new instance of the NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable class.
- List< Simulation > Run ()

  runs all simulations for a given rule

6.14.1 Beschreibung der Konstruktoren und Destruktoren

6.14.1.1 NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable.OnlyTestscenarioRunnable (Rule *Rule,* Network *N*)

Initializes a new instance of the NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable class.

Parameter

Rule	Rule object
N	Network

6.14.2 Dokumentation der Elementfunktionen

6.14.2.1 List<Simulation> NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable.Run ( )

runs all simulations for a given rule

Rückgabe

simulations that failed

Implementiert NSA.Model.BusinessLogic.TestscenarioRunnables.ITestscenarioRunnable.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↩ Logic/TestscenarioRunnables/OnlyTestscenarioRunnable.cs

6.15 NSA.Model.BusinessLogic.Packet Klassenreferenz

Class for a packet

Öffentliche Methoden

- Packet (Hardwarenode Src, Hardwarenode Dest, int T, bool ExpRes)
  Constructor for Packet
- Packet Send ()
  Sends this packet to the destination.

Propertys

- Hardwarenode Source `[get]`
  Returns the Source-Node
- Hardwarenode Destination `[get]`
  Returns the Destination-Node
- List< Hardwarenode > Hops `[get]`
  Returns the Hops of the packet
- int Ttl = new List<Hardwarenode>() `[get]`
- Result Result `[get]`
  Returns the result of the packet
- bool ExpectedResult = new Result() `[get]`
  Returns the expected result.

6.15.1 Ausführliche Beschreibung

Class for a packet

6.15.2 Beschreibung der Konstruktoren und Destruktoren

6.15.2.1 NSA.Model.BusinessLogic.Packet.Packet (Hardwarenode *Src,* Hardwarenode *Dest,* int *T,* bool *ExpRes*)

Constructor for Packet

Parameter

Src	Source-Node
Dest	Destination-Node
T	TTL
ExpRes	Expected Result

6.15.3 Dokumentation der Elementfunktionen

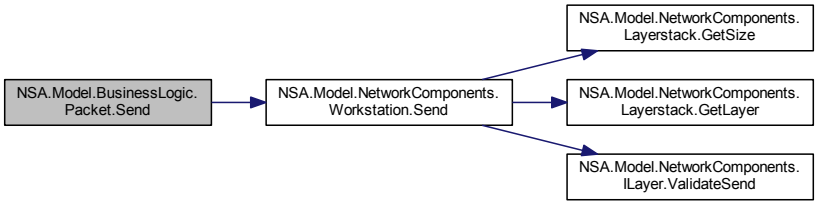
6.15.3.1 Packet NSA.Model.BusinessLogic.Packet.Send ( )

Sends this packet to the destination.

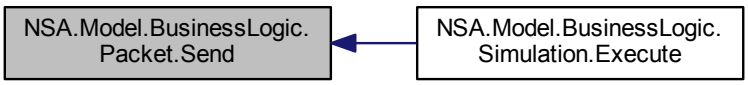
Rückgabe

The Returnpacket if sending to destination was successfull or null

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



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



6.15.4 Dokumentation der Propertys

6.15.4.1 Hardwarenode NSA.Model.BusinessLogic.Packet.Destination [get]

Returns the Destination-Node

The destination.

6.15.4.2 bool NSA.Model.BusinessLogic.Packet.ExpectedResult = new Result() [get]

Returns the expected result.

Expected Result

6.15.4.3 List<Hardwarenode> NSA.Model.BusinessLogic.Packet.Hops [get]

Returns the Hops of the packet

The Hops

6.15.4.4 Result NSA.Model.BusinessLogic.Packet.Result [get]

Returns the result of the packet

Result of the packet

6.15.4.5 Hardwarenode NSA.Model.BusinessLogic.Packet.Source [get]

Returns the Source-Node

The source

6.15.4.6 int NSA.Model.BusinessLogic.Packet.Ttl = new List<Hardwarenode>() [get]

The Time-To-Life

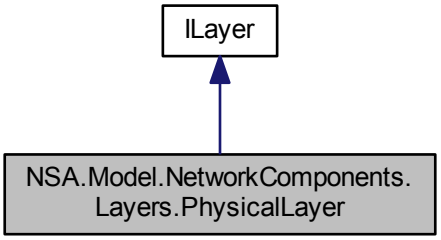
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↩ Logic/Packet.cs

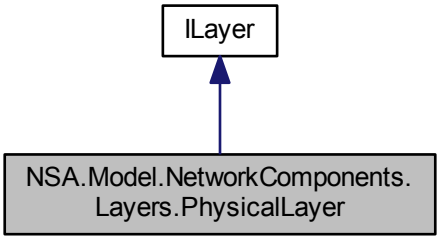
6.16 NSA.Model.NetworkComponents.Layers.PhysicalLayer Klassenreferenz

Physical-Layer

Klassendiagramm für NSA.Model.NetworkComponents.Layers.PhysicalLayer:



Zusammengehörigkeiten von NSA.Model.NetworkComponents.Layers.PhysicalLayer:



Öffentliche Methoden

- PhysicalLayer (int I)

  Initializes a new instance of the PhysicalLayer class.
- bool ValidateReceive (Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)

  Validates the layer while receiving a packet.
- string GetLayerName ()

  Gets the name of the layer.
- bool SetLayerName (string NewName)

  Sets the name of the layer.
- int GetLayerIndex ()

  Gets the index of the layer.
- void SetLayerIndex (int I)

Sets the index of the layer.

- void ValidateSend (Workstation Destination, Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, int LayerIndex)

  Validates the layer while sending a packet.

6.16.1 Ausführliche Beschreibung

Physical-Layer

Siehe auch

NSA.Model.NetworkComponents.ILayer

6.16.2 Beschreibung der Konstruktoren und Destruktoren

6.16.2.1 NSA.Model.NetworkComponents.Layers.PhysicalLayer.PhysicalLayer (int *l*)

Initializes a new instance of the PhysicalLayer class.

Parameter

l	The index.

6.16.3 Dokumentation der Elementfunktionen

6.16.3.1 int NSA.Model.NetworkComponents.Layers.PhysicalLayer.GetLayerIndex ( )

Gets the index of the layer.

Rückgabe

The index

Implementiert NSA.Model.NetworkComponents.ILayer.

6.16.3.2 string NSA.Model.NetworkComponents.Layers.PhysicalLayer.GetLayerName ( )

Gets the name of the layer.

Rückgabe

The Layername

Implementiert NSA.Model.NetworkComponents.ILayer.

6.16.3.3 void NSA.Model.NetworkComponents.Layers.PhysicalLayer.SetLayerIndex (int *l*)

Sets the index of the layer.

Parameter

I	The Index.

Implementiert NSA.Model.NetworkComponents.ILayer.

6.16.3.4 bool NSA.Model.NetworkComponents.Layers.PhysicalLayer.SetLayerName (string *NewName*)

Sets the name of the layer.

Parameter

NewName	New Name

Rückgabe

Implementiert NSA.Model.NetworkComponents.ILayer.

6.16.3.5 bool NSA.Model.NetworkComponents.Layers.PhysicalLayer.ValidateReceive (Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* Hardwarenode *Destination,* int *LayerIndex*)

Validates the layer while receiving a packet.

Parameter

CurrentNode	Current node
ValInfo	Validation Info
Tags	Tags
Destination	Destinationnode
LayerIndex	Index of the Layer

Rückgabe

Boolean value indicating if the validation was successfull

Implementiert NSA.Model.NetworkComponents.ILayer.

6.16.3.6 void NSA.Model.NetworkComponents.Layers.PhysicalLayer.ValidateSend (Workstation *Destination,* Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* int *LayerIndex*)

Validates the layer while sending a packet.

Parameter

Destination	The Destination

Parameter

CurrentNode	Current Node
ValInfo	Validation Info
Tags	Tags
LayerIndex	The Layer index

Implementiert NSA.Model.NetworkComponents.ILayer.

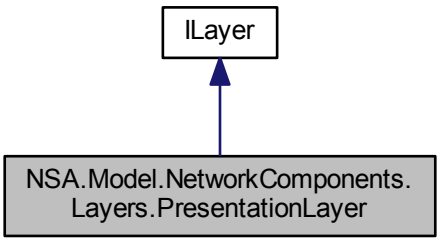
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩ Components/Layers/PhysicalLayer.cs

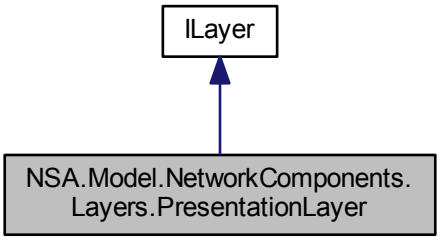
6.17 NSA.Model.NetworkComponents.Layers.PresentationLayer Klassenreferenz

Presentation-Layer

Klassendiagramm für NSA.Model.NetworkComponents.Layers.PresentationLayer:



Zusammengehörigkeiten von NSA.Model.NetworkComponents.Layers.PresentationLayer:



Öffentliche Methoden

- PresentationLayer (int I)

  Initializes a new instance of the PresentationLayer class.
- bool ValidateReceive (Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)

  Validates the layer while receiving a packet.
- string GetLayerName ()

  Gets the name of the layer.
- bool SetLayerName (string NewName)

  Sets the name of the layer.
- int GetLayerIndex ()

  Gets the index of the layer.
- void SetLayerIndex (int I)

  Sets the index of the layer.
- void ValidateSend (Workstation Destination, Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, int LayerIndex)

  Validates the layer while sending a packet.

6.17.1 Ausführliche Beschreibung

Presentation-Layer

Siehe auch

NSA.Model.NetworkComponents.ILayer

6.17.2 Beschreibung der Konstruktoren und Destruktoren

6.17.2.1 NSA.Model.NetworkComponents.Layers.PresentationLayer.PresentationLayer (int *I*)

Initializes a new instance of the PresentationLayer class.

Parameter

I	The index.

6.17.3 Dokumentation der Elementfunktionen

6.17.3.1 int NSA.Model.NetworkComponents.Layers.PresentationLayer.GetLayerIndex ( )

Gets the index of the layer.

Rückgabe

Index

Implementiert NSA.Model.NetworkComponents.ILayer.

6.17.3.2 string NSA.Model.NetworkComponents.Layers.PresentationLayer.GetLayerName ( )

Gets the name of the layer.

Rückgabe

The Layername

Implementiert NSA.Model.NetworkComponents.ILayer.

6.17.3.3 void NSA.Model.NetworkComponents.Layers.PresentationLayer.SetLayerIndex (int *I*)

Sets the index of the layer.

Parameter

I	The Index.

Implementiert NSA.Model.NetworkComponents.ILayer.

6.17.3.4 bool NSA.Model.NetworkComponents.Layers.PresentationLayer.SetLayerName (string *NewName*)

Sets the name of the layer.

Parameter

NewName	New Name

Rückgabe

Implementiert NSA.Model.NetworkComponents.ILayer.

6.17.3.5 bool NSA.Model.NetworkComponents.Layers.PresentationLayer.ValidateReceive (Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* Hardwarenode *Destination,* int *LayerIndex*)

Validates the layer while receiving a packet.

Parameter

CurrentNode	Current node
ValInfo	Validation Info
Tags	Tags
Destination	Destinationnode
LayerIndex	Index of the Layer

Rückgabe

Boolean value indicating if the validation was successfull

Implementiert NSA.Model.NetworkComponents.ILayer.

6.17.3.6 void NSA.Model.NetworkComponents.Layers.PresentationLayer.ValidateSend (Workstation *Destination,* Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* int *LayerIndex*)

Validates the layer while sending a packet.

Parameter

Destination	The Destination
CurrentNode	Current Node
ValInfo	Validation Info
Tags	Tags
LayerIndex	The Layer index

Implementiert NSA.Model.NetworkComponents.ILayer.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩ Components/Layers/PresentationLayer.cs

6.18 NSA.Model.BusinessLogic.Project Klassenreferenz

Class for project

Öffentliche Methoden

- Project ()

  Initializes a new instance of the Project class.

Propertys

- string Path `[get, set]`

  Gets or sets the path.
- Network Network `[get, set]`

  Gets or sets the network.

6.18.1 Ausführliche Beschreibung

Class for project

6.18.2 Beschreibung der Konstruktoren und Destruktoren

6.18.2.1 NSA.Model.BusinessLogic.Project.Project ( )

Initializes a new instance of the Project class.

6.18.3 Dokumentation der Propertys

6.18.3.1 Network NSA.Model.BusinessLogic.Project.Network [get], [set]

Gets or sets the network.

The network.

6.18.3.2 string NSA.Model.BusinessLogic.Project.Path [get], [set]

Gets or sets the path.

The path.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↩ Logic/Project.cs

6.19 NSA.Model.NetworkComponents.Helper_Classes.Result Klassenreferenz

Result class for the packetresult

Öffentliche Typen

- enum Errors {
  Errors.NoError, Errors.NoRoute, Errors.NoConnection, Errors.PacketNotForThisNode,
  Errors.SwitchNoConnection, Errors.SourceDestinationNull, Errors.TtlError, Errors.CustomLayerError,
  Errors.CustomLayerIndexError, Errors.NoPackets }

  Enum for the possible errors

Öffentliche Methoden

- Result ()

  Default constructor
- Result (Errors ErrId, string R, ILayer L)

  Constructor

Statische öffentliche Attribute

- static readonly string[] ResultStrings

  The possible result strings

Propertys

- Errors ErrorId `[get, set]`

  Gets or sets the error identifier.
- string Res `[get, set]`

  Gets or sets the result string.
- ILayer LayerError `[get, set]`

  Gets or sets the layer of the error.
- bool SendError `[get, set]`

  Gets or sets a value indicating whether it's a send error.

6.19.1 Ausführliche Beschreibung

Result class for the packetresult

6.19.2 Dokumentation der Aufzählungstypen

6.19.2.1 enum NSA.Model.NetworkComponents.Helper_Classes.Result.Errors `[strong]`

Enum for the possible errors

Aufzählungswerte

NoError No error

NoRoute No Route or Standard-Gateway was found

NoConnection No connection to the next node

PacketNotForThisNode The packet was not for this node

SwitchNoConnection The switch has no connection to the next node

SourceDestinationNull The source or destination is null

TtlError The TTL is zero

CustomLayerError Custom Layer Error

CustomLayerIndexError Custom layer index error

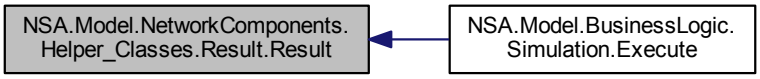
NoPackets No Packets in the simulation

6.19.3 Beschreibung der Konstruktoren und Destruktoren

6.19.3.1 NSA.Model.NetworkComponents.Helper_Classes.Result.Result ( )

Default constructor

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



6.19.3.2 NSA.Model.NetworkComponents.Helper_Classes.Result.Result (Errors *ErrId,* string *R,* ILayer *L*)

Constructor

Parameter

ErrId	The error identifier.
R	The result string.
L	The layer.

6.19.4 Dokumentation der Datenelemente

6.19.4.1 readonly string [ ] NSA.Model.NetworkComponents.Helper_Classes.Result.ResultStrings [static]

Initialisierung:

```
= { "Es ist kein Fehler bei der Simulation aufgetreten.",
            "Es gibt keine Route oder Standard-Gateway zum Zielrechner.",
            "Es gibt keine Verbindung zum nächsten Rechner.",
            "Das Packet war nicht für diesen Rechner bestimmt.",
            "Es gibt keine Verbindung zum nächsten Rechner.",
            "Quell- oder Zielrechner ist null.",
            "TTL ist 0, aber der Zielrechner wurde nicht erreicht.",
            "Layer {0} ist am Quell- aber nicht am Zielrechner enthalten.",
            "Layer {0} ist am Zielrechner am falschen Index enthalten.",
            "Keine Packete in der Simulation."
        }
```

The possible result strings

6.19.5 Dokumentation der Propertys

6.19.5.1 Errors NSA.Model.NetworkComponents.Helper_Classes.Result.ErrorId [get], [set]

Gets or sets the error identifier.

The error identifier.

6.19.5.2 ILayer NSA.Model.NetworkComponents.Helper_Classes.Result.LayerError [get], [set]

Gets or sets the layer of the error.

The layer of the error.

6.19.5.3 string NSA.Model.NetworkComponents.Helper_Classes.Result.Res [get], [set]

Gets or sets the result string.

The result string

6.19.5.4 bool NSA.Model.NetworkComponents.Helper_Classes.Result.SendError [get], [set]

Gets or sets a value indicating whether it's a send error.

`true` if send error otherwise, `false`.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩ Components/Helper Classes/Result.cs

6.20 NSA.Model.NetworkComponents.Route Klassenreferenz

class for a single route of the routingtable

Öffentliche Methoden

- Route (IPAddress Destination, IPAddress Subnetmask, IPAddress Gateway, Interface Iface)

  Initializes a new instance of the Route class.
- void SetRoute (IPAddress DestinationIp, IPAddress Mask, IPAddress GatewayAddress, Interface Intface)

  Sets the route.

Propertys

- string Name [get]

  Gets the name (id) of the route. Every route has a unique id.
- IPAddress Destination [get]

  Gets the destination IP.
- IPAddress Subnetmask [get]

  Gets the subnetmask.
- IPAddress Gateway [get]

  Gets the gateway.
- Interface Iface [get]

  Gets the interface.

6.20.1 Ausführliche Beschreibung

class for a single route of the routingtable

6.20.2 Beschreibung der Konstruktoren und Destruktoren

6.20.2.1 NSA.Model.NetworkComponents.Route.Route (IPAddress *Destination,* IPAddress *Subnetmask,* IPAddress *Gateway,* Interface *Iface*)

Initializes a new instance of the Route class.

Parameter

Destination	The Destination IP.
Subnetmask	The Mask.
Gateway	The Gateway.
Iface	The Interface to be used.

6.20.3 Dokumentation der Elementfunktionen

6.20.3.1 void NSA.Model.NetworkComponents.Route.SetRoute (IPAddress *DestinationIp,* IPAddress *Mask,* IPAddress *GatewayAddress,* Interface *Intface*)

Sets the route.

Parameter

DestinationIp	The destination.
Mask	The subnetmask.
GatewayAddress	The gateway.
Intface	The iface.

6.20.4 Dokumentation der Propertys

6.20.4.1 IPAddress NSA.Model.NetworkComponents.Route.Destination [get]

Gets the destination IP.

The destination.

6.20.4.2 IPAddress NSA.Model.NetworkComponents.Route.Gateway [get]

Gets the gateway.

The gateway.

6.20.4.3 Interface NSA.Model.NetworkComponents.Route.Iface [get]

Gets the interface.

The interface.

6.20.4.4 string NSA.Model.NetworkComponents.Route.Name [get]

Gets the name (id) of the route. Every route has a unique id.

The name.

6.20.4.5 IPAddress NSA.Model.NetworkComponents.Route.Subnetmask [get]

Gets the subnetmask.

The subnetmask.

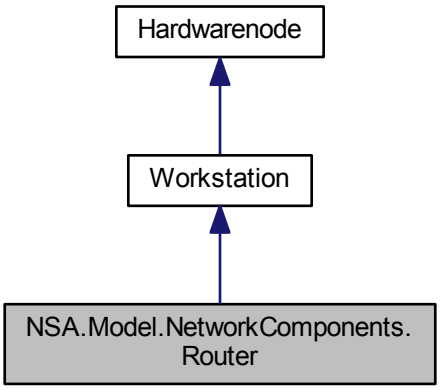
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩ Components/Route.cs

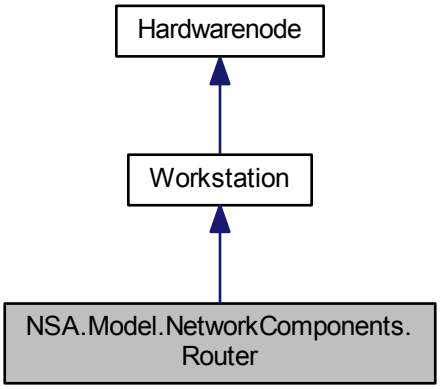
6.21 NSA.Model.NetworkComponents.Router Klassenreferenz

Implements the network component Router.

Klassendiagramm für NSA.Model.NetworkComponents.Router:



Zusammengehörigkeiten von NSA.Model.NetworkComponents.Router:



Öffentliche Methoden

- Router (string Name)

  Initializes a new instance of the Router class.

Propertys

- bool IsGateway `[get, set]`

  Gets or sets a value indicating whether this instance is an internet gateway.

Weitere Geerbte Elemente

6.21.1 Ausführliche Beschreibung

Implements the network component Router.

Siehe auch

NSA.Model.NetworkComponents.Workstation

6.21.2 Beschreibung der Konstruktoren und Destruktoren

6.21.2.1 NSA.Model.NetworkComponents.Router.Router (string *Name*)

Initializes a new instance of the Router class.

Parameter

Name	The name.

6.21.3 Dokumentation der Propertys

6.21.3.1 bool NSA.Model.NetworkComponents.Router.IsGateway `[get],[set]`

Gets or sets a value indicating whether this instance is an internet gateway.

`true` if this instance is an internet gateway; otherwise, `false`.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩ Components/Router.cs

6.22 NSA.Model.BusinessLogic.Rule Klassenreferenz

Rule class parses the text and if it is valid, the Rule object will be used for simulations

Öffentliche Methoden

- Rule (string StartNode, List< string > EndNodes, Dictionary< string, int > Options, SimulationType SimulationType, bool ExpectedResult, Network N)

  Initializes a new instance of the NSA.Model.BusinessLogic.Rule class.

Öffentliche, statische Methoden

- static Rule Parse (string Rule, Network N)

  parses a string and creates a rule from it
- static bool CheckForTrueOrFalse (string Text, string Rule)

  Checks if text contains TRUE or FALSE

Statische öffentliche Attribute

- static List< string > Parameters

  Accepted parameter types

Propertys

- SimulationType SimulType `[get]`

  Gets the type of the simul.
- string StartNodeString `[get]`

  String representation of startNode
- Hardwarenode StartNode `[get]`

  Gets the start node.
- List< string > EndNodesString `[get]`

  String representation of endNodes
- List< Hardwarenode > EndNodes `[get]`

  Gets the end nodes.
- Dictionary< string, int > Options `[get]`

  Rule options
- bool ExpectedResult `[get]`

  Expected result for the Rule

6.22.1 Ausführliche Beschreibung

Rule class parses the text and if it is valid, the Rule object will be used for simulations

Expected input: PC_NAME | (PC_NAME, ...) | { OPTIONS } | TRUE/FALSE PC_NAME | (SUBNET(PC_NAME), ...) | {TTL: 64, SSL: TRUE, ...} | TRUE/FALSE PC_NAME | ONLY(PC_NAME, ...) | {TTL: 64, SSL: TRUE, ...} | TRUE/FALSE PC_NAME | HAS_INTERNET | TRUE/FALSE

6.22.2 Beschreibung der Konstruktoren und Destruktoren

6.22.2.1 NSA.Model.BusinessLogic.Rule.Rule (string *StartNode,* List< string > *EndNodes,* Dictionary< string, int > *Options,* SimulationType *SimulationType,* bool *ExpectedResult,* Network *N*)

Initializes a new instance of the NSA.Model.BusinessLogic.Rule class.

Parameter

StartNode	Start node.
EndNodes	End nodes.
Options	Options.
SimulationType	Simulation type.
ExpectedResult	the expected result: True or False
N	Network

6.22.3 Dokumentation der Elementfunktionen

6.22.3.1 static bool NSA.Model.BusinessLogic.Rule.CheckForTrueOrFalse (string *Text,* string *Rule*) [static]

Checks if text contains TRUE or FALSE

Rückgabe

true if TRUE, false if FALSE, else throws an exception

Parameter

Text	text to parse
Rule	Rule string, which is needed, in order to inform the user about invalid Rule input

6.22.3.2 static Rule NSA.Model.BusinessLogic.Rule.Parse (string *Rule,* Network *N*) [static]

parses a string and creates a rule from it

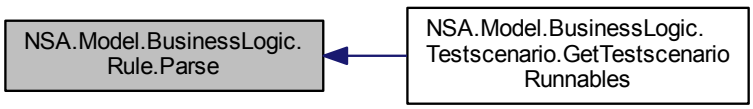
Parameter

Rule	
N	

Rückgabe

Rule object

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



6.22.4 Dokumentation der Datenelemente

6.22.4.1 List<string> NSA.Model.BusinessLogic.Rule.Parameters [static]

Initialisierung:

```
= new List<string>
        {
            "TTL",
            "SSL"
        }
```

Accepted parameter types

6.22.5 Dokumentation der Propertys

6.22.5.1 List<Hardwarenode> NSA.Model.BusinessLogic.Rule.EndNodes [get]

Gets the end nodes.

6.22.5.2 List<string> NSA.Model.BusinessLogic.Rule.EndNodesString [get]

String representation of endNodes

6.22.5.3 bool NSA.Model.BusinessLogic.Rule.ExpectedResult [get]

Expected result for the Rule

6.22.5.4 Dictionary<string, int> NSA.Model.BusinessLogic.Rule.Options [get]

Rule options

6.22.5.5 SimulationType NSA.Model.BusinessLogic.Rule.SimulType [get]

Gets the type of the simul.

The type of the simul.

6.22.5.6 Hardwarenode NSA.Model.BusinessLogic.Rule.StartNode [get]

Gets the start node.

6.22.5.7 string NSA.Model.BusinessLogic.Rule.StartNodeString [get]

String representation of startNode

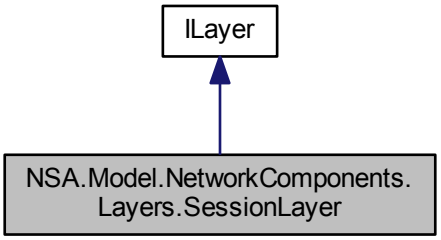
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/BusinessLogic/Rule.cs

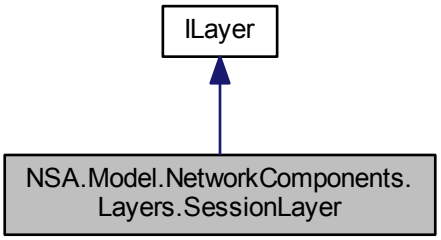
6.23 NSA.Model.NetworkComponents.Layers.SessionLayer Klassenreferenz

Session-Layer

Klassendiagramm für NSA.Model.NetworkComponents.Layers.SessionLayer:



Zusammengehörigkeiten von NSA.Model.NetworkComponents.Layers.SessionLayer:



Öffentliche Methoden

- SessionLayer (int l)

  Initializes a new instance of the SessionLayer class.
- bool ValidateReceive (Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)

  Validates the layer while receiving a packet.
- string GetLayerName ()

  Gets the name of the layer.
- bool SetLayerName (string NewName)

  Sets the name of the layer.
- int GetLayerIndex ()

  Gets the index of the layer.
- void SetLayerIndex (int l)

  Sets the index of the layer.
- void ValidateSend (Workstation Destination, Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, int LayerIndex)

  Validates the layer while sending a packet.

6.23.1 Ausführliche Beschreibung

Session-Layer

Siehe auch

NSA.Model.NetworkComponents.ILayer

6.23.2 Beschreibung der Konstruktoren und Destruktoren

6.23.2.1 NSA.Model.NetworkComponents.Layers.SessionLayer.SessionLayer (int *l*)

Initializes a new instance of the SessionLayer class.

Parameter

l	The index.

6.23.3 Dokumentation der Elementfunktionen

6.23.3.1 int NSA.Model.NetworkComponents.Layers.SessionLayer.GetLayerIndex ( )

Gets the index of the layer.

Rückgabe

Index

Implementiert NSA.Model.NetworkComponents.ILayer.

6.23.3.2 string NSA.Model.NetworkComponents.Layers.SessionLayer.GetLayerName ( )

Gets the name of the layer.

Rückgabe

The Layername

Implementiert NSA.Model.NetworkComponents.ILayer.

6.23.3.3 void NSA.Model.NetworkComponents.Layers.SessionLayer.SetLayerIndex (int *I*)

Sets the index of the layer.

Parameter

I	The Index.

Implementiert NSA.Model.NetworkComponents.ILayer.

6.23.3.4 bool NSA.Model.NetworkComponents.Layers.SessionLayer.SetLayerName (string *NewName*)

Sets the name of the layer.

Parameter

NewName	New Name

Rückgabe

Implementiert NSA.Model.NetworkComponents.ILayer.

6.23.3.5 bool NSA.Model.NetworkComponents.Layers.SessionLayer.ValidateReceive (Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* Hardwarenode *Destination,* int *LayerIndex*)

Validates the layer while receiving a packet.

Parameter

CurrentNode	Current node
ValInfo	Validation Info
Tags	Tags
Destination	Destinationnode
LayerIndex	Index of the Layer

Rückgabe

Boolean value indicating if the validation was successfull

Implementiert NSA.Model.NetworkComponents.ILayer.

6.23.3.6 void NSA.Model.NetworkComponents.Layers.SessionLayer.ValidateSend (Workstation *Destination,* Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* int *LayerIndex*)

Validates the layer while sending a packet.

Parameter

Destination	The Destination
CurrentNode	Current Node
ValInfo	Validation Info
Tags	Tags
LayerIndex	The Layer index

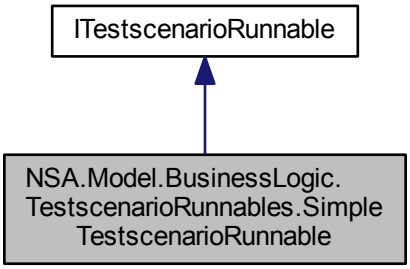
Implementiert NSA.Model.NetworkComponents.ILayer.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

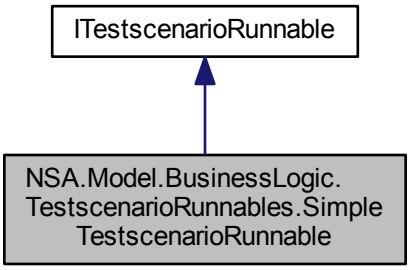
- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩ Components/Layers/SessionLayer.cs

6.24 NSA.Model.BusinessLogic.TestscenarioRunnables.SimpleTestscenarioRunnable Klassenreferenz

Klassendiagramm für NSA.Model.BusinessLogic.TestscenarioRunnables.SimpleTestscenarioRunnable:



Zusammengehörigkeiten von NSA.Model.BusinessLogic.TestscenarioRunnables.SimpleTestscenarioRunnable:



Öffentliche Methoden

- SimpleTestscenarioRunnable (Rule Rule)

  Initializes a new instance of the NSA.Model.BusinessLogic.TestscenarioRunnables.SimpleTestscenarioRunnable class.
- List< Simulation > Run ()

  runs all simulations for a given rule

6.24.1 Beschreibung der Konstruktoren und Destruktoren

6.24.1.1 NSA.Model.BusinessLogic.TestscenarioRunnables.SimpleTestscenarioRunnable.SimpleTestscenarioRunnable (Rule *Rule*)

Initializes a new instance of the NSA.Model.BusinessLogic.TestscenarioRunnables.SimpleTestscenarioRunnable class.

Parameter

Rule	Rule object

6.24.2 Dokumentation der Elementfunktionen

6.24.2.1 List<Simulation> NSA.Model.BusinessLogic.TestscenarioRunnables.SimpleTestscenarioRunnable.Run ( )

runs all simulations for a given rule

Rückgabe

simulations that failed

Implementiert NSA.Model.BusinessLogic.TestscenarioRunnables.ITestscenarioRunnable.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↩ Logic/TestscenarioRunnables/SimpleTestscenarioRunnable.cs

6.25 NSA.Model.BusinessLogic.Simulation Klassenreferenz

Class for the simulation

Öffentliche Methoden

- Simulation (string I)
  Constructor for the simultion used in testscenarios
- Simulation (string I, string S, string D, bool ExpRes)
  Constructor used for normal simulation
- void AddPacketSend (Packet Packet)
  Adds a packet to the sendpackets
- Result Execute ()
  Executes this simulation
- IEnumerable< Packet > GetAllPackets ()
  Returns all packets
- Packet GetLastPacket ()
  Returns the last packet

Propertys

- List< Packet > PacketsSend `[get]`
  Returns the sendpackets
- List< Packet > PacketsReceived = new List<Packet>() `[get]`
  Returns the receivedpackets
- string Source = new List<Packet>() `[get]`
  Returns the name of the sourcenode
- string Destination `[get]`
  Returns the name of the destinationnode
- string Id `[get]`
  Returns the ID
- bool ExpectedResult `[get]`
  Returns the expected result

6.25.1 Ausführliche Beschreibung

Class for the simulation

6.25.2 Beschreibung der Konstruktoren und Destruktoren

6.25.2.1 NSA.Model.BusinessLogic.Simulation.Simulation (string *I*)

Constructor for the simultion used in testscenarios

Parameter

I	The ID

6.25.2.2 NSA.Model.BusinessLogic.Simulation.Simulation (string *I*, string *S*, string *D*, bool *ExpRes*)

Constructor used for normal simulation

Parameter

I	ID
S	Name of sourcenode
D	Name of destinationnode
ExpRes	Expected result

6.25.3 Dokumentation der Elementfunktionen

6.25.3.1 void NSA.Model.BusinessLogic.Simulation.AddPacketSend (Packet *Packet*)

Adds a packet to the sendpackets

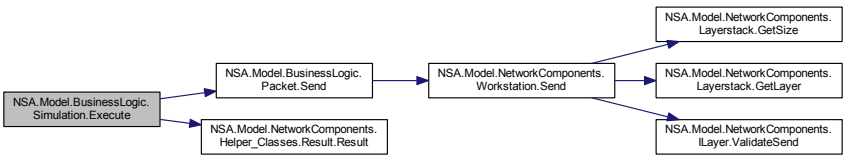
Parameter

Packet	The packet to be added

6.25.3.2 Result NSA.Model.BusinessLogic.Simulation.Execute ( )

Executes this simulation

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



6.25.3.3 IEnumerable<Packet> NSA.Model.BusinessLogic.Simulation.GetAllPackets ( )

Returns all packets

Rückgabe

All Packets

6.25.3.4 Packet NSA.Model.BusinessLogic.Simulation.GetLastPacket ( )

Returns the last packet

Rückgabe

Null (if there is no packet) or the last packet

6.25.4 Dokumentation der Propertys

6.25.4.1 string NSA.Model.BusinessLogic.Simulation.Destination [get]

Returns the name of the destinationnode

Name of the destinationnode

6.25.4.2 bool NSA.Model.BusinessLogic.Simulation.ExpectedResult [get]

Returns the expected result

The expected result

6.25.4.3 string NSA.Model.BusinessLogic.Simulation.Id [get]

Returns the ID

ID of the simulation

6.25.4.4 List<Packet> NSA.Model.BusinessLogic.Simulation.PacketsReceived = new List<Packet>() [get]

Returns the receivedpackets

The receivedpackets

6.25.4.5 List<Packet> NSA.Model.BusinessLogic.Simulation.PacketsSend [get]

Returns the sendpackets

The sendpackets

6.25.4.6 string NSA.Model.BusinessLogic.Simulation.Source = new List<Packet>() [get]

Returns the name of the sourcenode

Name of the sourcenode

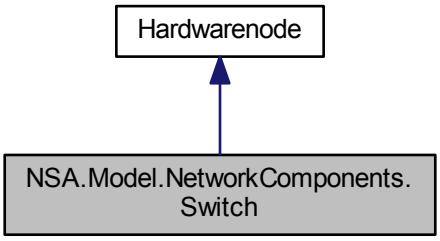
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↩ Logic/Simulation.cs

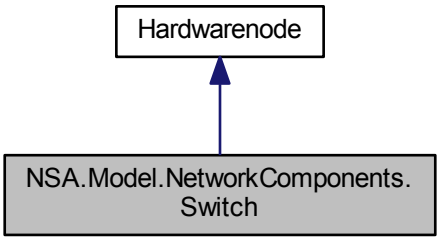
6.26 NSA.Model.NetworkComponents.Switch Klassenreferenz

Implements the network component switch.

Klassendiagramm für NSA.Model.NetworkComponents.Switch:



Zusammengehörigkeiten von NSA.Model.NetworkComponents.Switch:



Öffentliche Methoden

- Switch (string Name)

  Initializes a new instance of the Switch class.
- override Interface AddInterface (IPAddress Ip, IPAddress Subnetmask, int PortNum=-1)

  Adds a new interface with the given IP and subnetmask
- List< Interface > SetInterfaceCount (int Count)

  Sets the interface count to the given value.
- override List< Hardwarenode > Send (Hardwarenode Destination, Dictionary< string, object > Tags, ValidationInfo ValInfo)

  Hardwarenode sends the package to specified destination.
- bool SendToIp (ValidationInfo ValInfo, Connection ComingConn)

  Sends to ip.
- bool SendToDestination (Workstation Destination, ValidationInfo ValInfo, Connection ComingCon, IPAddress nodeIP, IPAddress subnetmask)

  Sends to destination.

Weitere Geerbte Elemente

6.26.1 Ausführliche Beschreibung

Implements the network component switch.

Siehe auch

NSA.Model.NetworkComponents.Hardwarenode

6.26.2 Beschreibung der Konstruktoren und Destruktoren

6.26.2.1 NSA.Model.NetworkComponents.Switch.Switch (string *Name*)

Initializes a new instance of the Switch class.

Parameter

Name	The name of the switch.

6.26.3 Dokumentation der Elementfunktionen

6.26.3.1 override Interface NSA.Model.NetworkComponents.Switch.AddInterface (IPAddress *Ip,* IPAddress *Subnetmask,* int *PortNum* = `-1`) `[virtual]`

Adds a new interface with the given IP and subnetmask

Parameter

Ip	The IP of the interface. Ignored if used with switch
Subnetmask	The subnetmask. Ignored if used with switch
PortNum	Number of port. Only for project loading purpose.

Rückgabe

The newly added Interface

Erneute Implementation von NSA.Model.NetworkComponents.Hardwarenode.

6.26.3.2 override List<Hardwarenode> NSA.Model.NetworkComponents.Switch.Send (Hardwarenode *Destination,* Dictionary< string, object > *Tags,* ValidationInfo *ValInfo*) `[virtual]`

Hardwarenode sends the package to specified destination.

Parameter

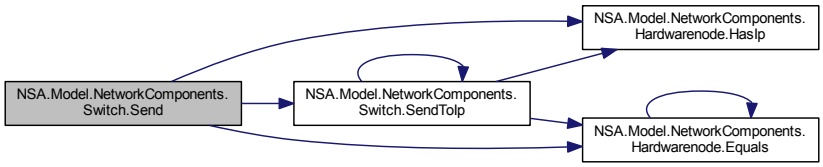
Destination	The destination.
Tags	Optional tags.
ValInfo	Validation Info

Rückgabe

The Hardwarenode which received the package or null if an error occured

Erneute Implementation von NSA.Model.NetworkComponents.Hardwarenode.

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



6.26.3.3 bool NSA.Model.NetworkComponents.Switch.SendToDestination (Workstation *Destination,* ValidationInfo *ValInfo,* Connection *ComingCon,* IPAddress *nodeIP,* IPAddress *subnetmask*)

Sends to destination.

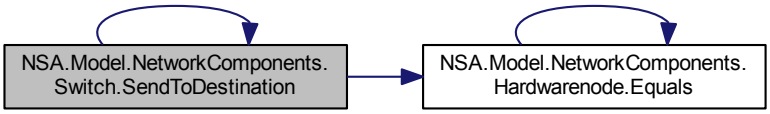
Parameter

Destination	The destination.
ValInfo	The value information.
ComingCon	The coming connection.
nodeIP	The node ip of the sending node.
subnetmask	The subnetmask.

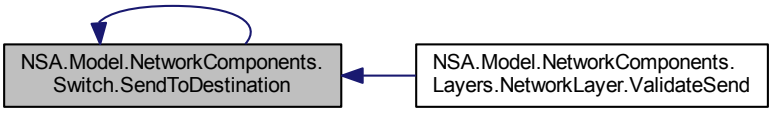
Rückgabe

Bool if it could send

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



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



6.26.3.4 bool NSA.Model.NetworkComponents.Switch.SendToIp (ValidationInfo *ValInfo,* Connection *ComingConn*)

Sends to ip.

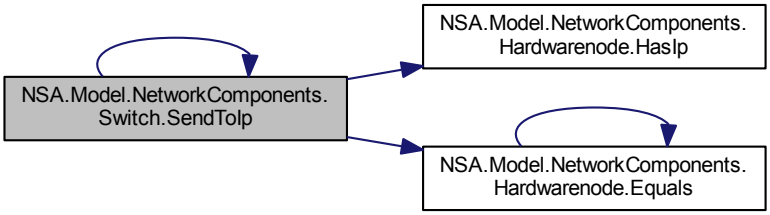
Parameter

ValInfo	The value information.
ComingConn	The coming connection.

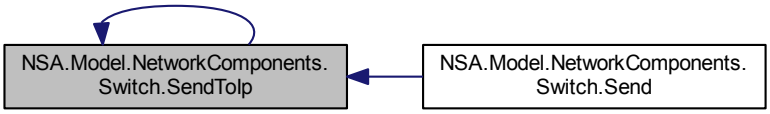
Rückgabe

Bool if it could send

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



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



6.26.3.5 List<Interface> NSA.Model.NetworkComponents.Switch.SetInterfaceCount (int *Count*)

Sets the interface count to the given value.

Parameter

Count	The count.

Rückgabe

Returns a list of removed interfaces

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩ Components/Switch.cs

6.27 NSA.Model.BusinessLogic.Testscenario Klassenreferenz

Öffentliche Methoden

- Testscenario (string T, Network N, string FileName)

  Initializes a new instance of the NSA.Model.BusinessLogic.Testscenario class.
- List< ITestscenarioRunnable > GetTestscenarioRunnables ()

  parses the text and creates Testscenario runnables

Propertys

- string FileName `[get]`

  Gets the name of the file.
- string Id `[get]`

  Gets the identifier.

6.27.1 Beschreibung der Konstruktoren und Destruktoren

6.27.1.1 NSA.Model.BusinessLogic.Testscenario.Testscenario (string *T,* Network *N,* string *FileName*)

Initializes a new instance of the NSA.Model.BusinessLogic.Testscenario class.

Parameter

T	Text to parse
N	Network
FileName	File name

6.27.2 Dokumentation der Elementfunktionen

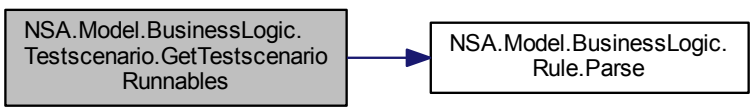
6.27.2.1 List<ITestscenarioRunnable> NSA.Model.BusinessLogic.Testscenario.GetTestscenarioRunnables ( )

parses the text and creates Testscenario runnables

Rückgabe

runnables

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



6.27.3 Dokumentation der Propertys

6.27.3.1 string NSA.Model.BusinessLogic.Testscenario.FileName [get]

Gets the name of the file.

The name of the file.

6.27.3.2 string NSA.Model.BusinessLogic.Testscenario.Id [get]

Gets the identifier.

The identifier.

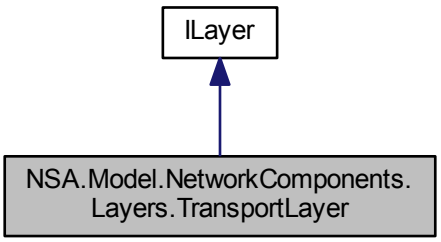
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↩ Logic/Testscenario.cs

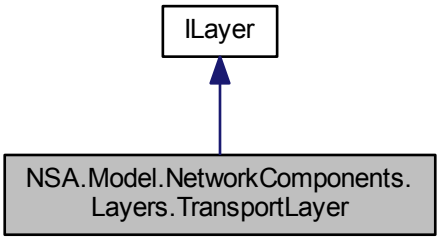
6.28 NSA.Model.NetworkComponents.Layers.TransportLayer Klassenreferenz

Transport-Layer

Klassendiagramm für NSA.Model.NetworkComponents.Layers.TransportLayer:



Zusammengehörigkeiten von NSA.Model.NetworkComponents.Layers.TransportLayer:



Öffentliche Methoden

- TransportLayer (int l)

  Initializes a new instance of the TransportLayer class.
- bool ValidateReceive (Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)

  Validates the layer while receiving a packet.
- string GetLayerName ()

  Gets the name of the layer.
- bool SetLayerName (string NewName)

  Sets the name of the layer.
- int GetLayerIndex ()

  Gets the index of the layer.
- void SetLayerIndex (int l)

  Sets the index of the layer.
- void ValidateSend (Workstation Destination, Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, int LayerIndex)

  Validates the layer while sending a packet.

6.28.1 Ausführliche Beschreibung

Transport-Layer

Siehe auch

NSA.Model.NetworkComponents.ILayer

6.28.2 Beschreibung der Konstruktoren und Destruktoren

6.28.2.1 NSA.Model.NetworkComponents.Layers.TransportLayer.TransportLayer (int *l*)

Initializes a new instance of the TransportLayer class.

Parameter

l	The i.

6.28.3 Dokumentation der Elementfunktionen

6.28.3.1 int NSA.Model.NetworkComponents.Layers.TransportLayer.GetLayerIndex ( )

Gets the index of the layer.

Rückgabe

Index

Implementiert NSA.Model.NetworkComponents.ILayer.

6.28.3.2 string NSA.Model.NetworkComponents.Layers.TransportLayer.GetLayerName ( )

Gets the name of the layer.

Rückgabe

The Layername

Implementiert NSA.Model.NetworkComponents.ILayer.

6.28.3.3 void NSA.Model.NetworkComponents.Layers.TransportLayer.SetLayerIndex (int *I*)

Sets the index of the layer.

Parameter

I	The Index.

Implementiert NSA.Model.NetworkComponents.ILayer.

6.28.3.4 bool NSA.Model.NetworkComponents.Layers.TransportLayer.SetLayerName (string *NewName*)

Sets the name of the layer.

Parameter

NewName	New Name

Rückgabe

Implementiert NSA.Model.NetworkComponents.ILayer.

6.28.3.5 bool NSA.Model.NetworkComponents.Layers.TransportLayer.ValidateReceive (Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* Hardwarenode *Destination,* int *LayerIndex*)

Validates the layer while receiving a packet.

Parameter

CurrentNode	Current node
ValInfo	Validation Info
Tags	Tags
Destination	Destinationnode
LayerIndex	Index of the Layer

Rückgabe

Boolean value indicating if the validation was successfull

Implementiert NSA.Model.NetworkComponents.ILayer.

6.28.3.6 void NSA.Model.NetworkComponents.Layers.TransportLayer.ValidateSend (Workstation *Destination,* Workstation *CurrentNode,* ValidationInfo *ValInfo,* Dictionary< string, object > *Tags,* int *LayerIndex*)

Validates the layer while sending a packet.

Parameter

Destination	The Destination
CurrentNode	Current Node
ValInfo	Validation Info
Tags	Tags
LayerIndex	The Layer index

Implementiert NSA.Model.NetworkComponents.ILayer.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩ Components/Layers/TransportLayer.cs

6.29 NSA.Model.NetworkComponents.Helper_Classes.ValidationInfo Klassenreferenz

Helper class for the parameters of the simulation

Propertys

- List< Hardwarenode > NextNodes `[get, set]`

  Gets or sets the next nodes.
- IPAddress NextNodeIp `[get, set]`

  Gets or sets the next node ip.
- Interface Iface `[get, set]`

  Gets or sets the interface.
- Result Res `[get, set]`

  Gets or sets the result.
- Workstation Source `[get, set]`

  Gets or sets the source.

6.29.1 Ausführliche Beschreibung

Helper class for the parameters of the simulation

6.29.2 Dokumentation der Propertys

6.29.2.1 Interface NSA.Model.NetworkComponents.Helper_Classes.ValidationInfo.Iface [get], [set]

Gets or sets the interface.

The interface.

6.29.2.2 IPAddress NSA.Model.NetworkComponents.Helper_Classes.ValidationInfo.NextNodeIp [get], [set]

Gets or sets the next node ip.

The next node ip.

6.29.2.3 List<Hardwarenode> NSA.Model.NetworkComponents.Helper_Classes.ValidationInfo.NextNodes [get], [set]

Gets or sets the next nodes.

The next nodes.

6.29.2.4 Result NSA.Model.NetworkComponents.Helper_Classes.ValidationInfo.Res [get], [set]

Gets or sets the result.

The result.

6.29.2.5 Workstation NSA.Model.NetworkComponents.Helper_Classes.ValidationInfo.Source [get], [set]

Gets or sets the source.

The source.

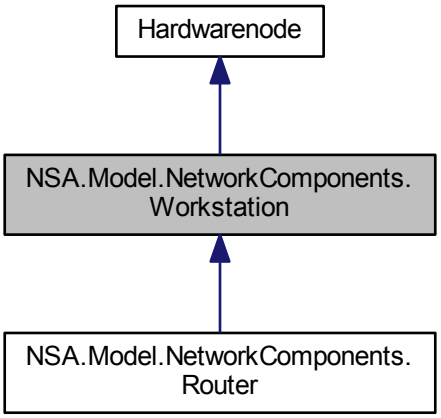
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↩ Components/Helper Classes/ValidationInfo.cs

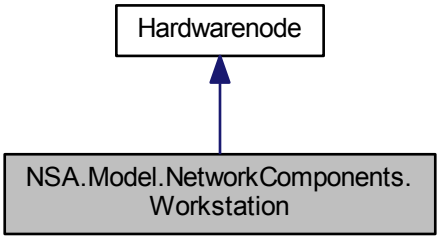
6.30 NSA.Model.NetworkComponents.Workstation Klassenreferenz

Implements the network component Workstation.

Klassendiagramm für NSA.Model.NetworkComponents.Workstation:



Zusammengehörigkeiten von NSA.Model.NetworkComponents.Workstation:



Öffentliche Methoden

- Workstation (string Name)

  Initializes a new instance of the Workstation class. The IP address of the standardgateway must be set seperatly.
- void AddRoute (Route Route)

  Adds the route.
- void RemoveRoute (string N)

  Removes the route.
- int GetRouteCount ()

Gets the route count.

- Dictionary< string, Route >.ValueCollection GetRoutes ()

  Gets the routes.
- Route GetRouteAt (int Index)

  Gets the route at the given index.
- bool SetRoute (string RouteName, IPAddress Destination, IPAddress Subnetmask, IPAddress Gateway, Interface Iface)

  Sets the route.
- override bool HasIp (IPAddress Ip)

  Checks if the Hardwarenode has the IP
- override List< Hardwarenode > Send (Hardwarenode Destination, Dictionary< string, object > Tags, ValidationInfo ValInfo)

  Hardwarenode sends the package to specified destination.
- override bool Receive (Dictionary< string, object > Tags, ValidationInfo ValInfo, Hardwarenode Destination)

  Hardwarenode receives the package.

Propertys

- IPAddress StandardGateway `[get, set]`

  Gets or sets the standard gateway.
- Interface StandardGatewayPort `[get, set]`

  Gets or sets the standard gateway port.

Weitere Geerbte Elemente

6.30.1 Ausführliche Beschreibung

Implements the network component Workstation.

Siehe auch

NSA.Model.NetworkComponents.Hardwarenode

6.30.2 Beschreibung der Konstruktoren und Destruktoren

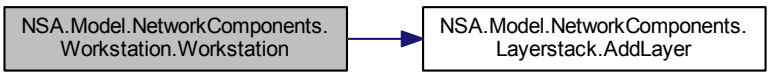
6.30.2.1 NSA.Model.NetworkComponents.Workstation.Workstation (string *Name*)

Initializes a new instance of the Workstation class. The IP address of the standardgateway must be set seperatly.

Parameter

Name	The Name.

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



6.30.3 Dokumentation der Elementfunktionen

6.30.3.1 void NSA.Model.NetworkComponents.Workstation.AddRoute (Route *Route*)

Adds the route.

Parameter

Route	The route.

6.30.3.2 Route NSA.Model.NetworkComponents.Workstation.GetRouteAt (int *Index*)

Gets the route at the given index.

Parameter

Index	The index.

Rückgabe

6.30.3.3 int NSA.Model.NetworkComponents.Workstation.GetRouteCount ( )

Gets the route count.

Rückgabe

int: number of routes in the routingtable

6.30.3.4 Dictionary<string, Route>.ValueCollection NSA.Model.NetworkComponents.Workstation.GetRoutes ( )

Gets the routes.

Rückgabe

The Routes

6.30.3.5 override bool NSA.Model.NetworkComponents.Workstation.HasIp (IPAddress *Ip*) [virtual]

Checks if the Hardwarenode has the IP

Parameter

Ip	The ip.

Rückgabe

bool: true if workstation has the ip, otherwise false

Erneute Implementation von NSA.Model.NetworkComponents.Hardwarenode.

6.30.3.6 override bool NSA.Model.NetworkComponents.Workstation.Receive (Dictionary< string, object > *Tags,* ValidationInfo *ValInfo,* Hardwarenode *Destination*) [virtual]

Hardwarenode receives the package.

Parameter

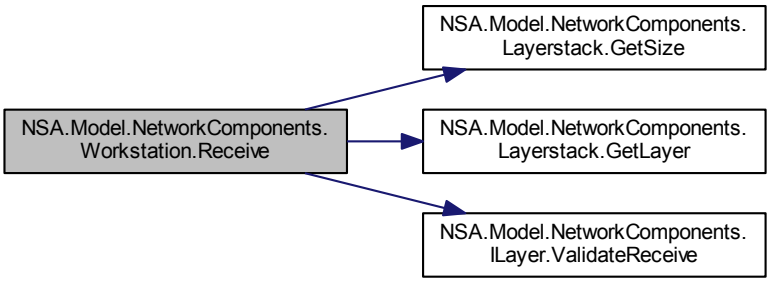
Tags	Optional tags.
ValInfo	The validation Info
Destination	The destination

Rückgabe

bool that indicates if the Hardwarenode received the package

Erneute Implementation von NSA.Model.NetworkComponents.Hardwarenode.

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



6.30.3.7 void NSA.Model.NetworkComponents.Workstation.RemoveRoute (string *N*)

Removes the route.

Parameter

N	The name.

6.30.3.8 override List<Hardwarenode> NSA.Model.NetworkComponents.Workstation.Send (Hardwarenode *Destination,* Dictionary< string, object > *Tags,* ValidationInfo *ValInfo*) [virtual]

Hardwarenode sends the package to specified destination.

Parameter

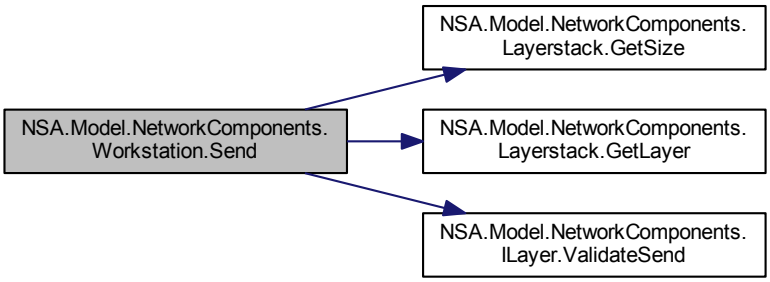
Destination	The destination.
Tags	Optional tags.
ValInfo	

Rückgabe

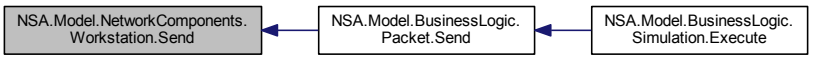
The Hardwarenode which received the package or null if an error occured

Erneute Implementation von NSA.Model.NetworkComponents.Hardwarenode.

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



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



6.30.3.9 bool NSA.Model.NetworkComponents.Workstation.SetRoute (string *RouteName,* IPAddress *Destination,* IPAddress *Subnetmask,* IPAddress *Gateway,* Interface *Iface*)

Sets the route.

Parameter

RouteName	The name of the route.
Destination	The new destination.
Subnetmask	The new subnetmask.
Gateway	The new gateway.
Iface	The new interface.

Rückgabe

bool: false if the route could not be found, otherwise true

6.30.4 Dokumentation der Propertys

6.30.4.1 IPAddress NSA.Model.NetworkComponents.Workstation.StandardGateway [get],[set]

Gets or sets the standard gateway.

The standard gateway.

6.30.4.2 Interface NSA.Model.NetworkComponents.Workstation.StandardGatewayPort [get],[set]

Gets or sets the standard gateway port.

The standard gateway port.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Workstation.cs

Kapitel 7

Datei-Dokumentation

7.1 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/BusinessLogic/Packet.cs-Dateireferenz

Klassen

- class NSA.Model.BusinessLogic.Packet

  Class for a packet

Namensbereiche

- namespace NSA.Model.BusinessLogic

7.2 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/BusinessLogic/Project.cs-Dateireferenz

Klassen

- class NSA.Model.BusinessLogic.Project

  Class for project

Namensbereiche

- namespace NSA.Model.BusinessLogic

7.3 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/BusinessLogic/Rule.cs-Dateireferenz

Klassen

- class NSA.Model.BusinessLogic.Rule

  Rule class parses the text and if it is valid, the Rule object will be used for simulations

Namensbereiche

- namespace NSA.Model.BusinessLogic

Aufzählungen

- enum NSA.Model.BusinessLogic.SimulationType { NSA.Model.BusinessLogic.SimulationType.Simple = 0, NSA.Model.BusinessLogic.SimulationType.Only = 1, NSA.Model.BusinessLogic.SimulationType.HasInternet = 2 }

  Simulation types

7.4 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/BusinessLogic/Simulation.cs-Dateireferenz

Klassen

- class NSA.Model.BusinessLogic.Simulation

  Class for the simulation

Namensbereiche

- namespace NSA.Model.BusinessLogic

7.5 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/BusinessLogic/Testscenario.cs-Dateireferenz

Klassen

- class NSA.Model.BusinessLogic.Testscenario

Namensbereiche

- namespace NSA.Model.BusinessLogic

7.6 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/BusinessLogic/TestscenarioRunnables/HasInternetTestscenarioRunnable.cs-Dateireferenz

Klassen

- class NSA.Model.BusinessLogic.TestscenarioRunnables.HasInternetTestscenarioRunnable

Namensbereiche

- namespace NSA.Model.BusinessLogic.TestscenarioRunnables

7.7 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/BusinessLogic/TestscenarioRunnables/ITestscenarioRunnable.cs-↩ Dateireferenz

Klassen

- interface NSA.Model.BusinessLogic.TestscenarioRunnables.ITestscenarioRunnable

Namensbereiche

- namespace NSA.Model.BusinessLogic.TestscenarioRunnables

7.8 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/BusinessLogic/TestscenarioRunnables/OnlyTestscenarioRunnable.cs-↩ Dateireferenz

Klassen

- class NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable

Namensbereiche

- namespace NSA.Model.BusinessLogic.TestscenarioRunnables

7.9 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/BusinessLogic/TestscenarioRunnables/SimpleTestscenarioRunnable.cs-Dateireferenz

Klassen

- class NSA.Model.BusinessLogic.TestscenarioRunnables.SimpleTestscenarioRunnable

Namensbereiche

- namespace NSA.Model.BusinessLogic.TestscenarioRunnables

7.10 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Connection.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Connection
  Class for a connection between two hardwarenodes.

Namensbereiche

- namespace NSA.Model.NetworkComponents

7.11 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Hardwarenode.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Hardwarenode
  Implements the basis class for hardwarenodes.

Namensbereiche

- namespace NSA.Model.NetworkComponents

7.12 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Helper Classes/IPAddressExtensions.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Helper_Classes.IPAddressExtensions

Namensbereiche

- namespace NSA.Model.NetworkComponents.Helper_Classes

7.13 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Helper Classes/Result.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Helper_Classes.Result
  Result class for the packetresult

Namensbereiche

- namespace NSA.Model.NetworkComponents.Helper_Classes

7.14 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/NetworkComponents/Helper Classes/ValidationInfo.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Helper_Classes.ValidationInfo

  Helper class for the parameters of the simulation

Namensbereiche

- namespace NSA.Model.NetworkComponents.Helper_Classes

7.15 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/NetworkComponents/ILayer.cs-Dateireferenz

Klassen

- interface NSA.Model.NetworkComponents.ILayer

Namensbereiche

- namespace NSA.Model.NetworkComponents

7.16 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/NetworkComponents/Interface.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Interface

  Class for a interface of an hardwarenode.

Namensbereiche

- namespace NSA.Model.NetworkComponents

7.17 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Layers/ApplicationLayer.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Layers.ApplicationLayer
  Application-Layer

Namensbereiche

- namespace NSA.Model.NetworkComponents.Layers

7.18 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Layers/CustomLayer.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Layers.CustomLayer
  Custom-Layer

Namensbereiche

- namespace NSA.Model.NetworkComponents.Layers

7.19 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Layers/DataLinkLayer.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Layers.DataLinkLayer
  DataLink-Layer

Namensbereiche

- namespace NSA.Model.NetworkComponents.Layers

7.20 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Layers/NetworkLayer.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Layers.NetworkLayer
  Network-Layer

Namensbereiche

- namespace NSA.Model.NetworkComponents.Layers

7.21 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/NetworkComponents/Layers/PhysicalLayer.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Layers.PhysicalLayer

  Physical-Layer

Namensbereiche

- namespace NSA.Model.NetworkComponents.Layers

7.22 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/NetworkComponents/Layers/PresentationLayer.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Layers.PresentationLayer

  Presentation-Layer

Namensbereiche

- namespace NSA.Model.NetworkComponents.Layers

7.23 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/NetworkComponents/Layers/SessionLayer.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Layers.SessionLayer

  Session-Layer

Namensbereiche

- namespace NSA.Model.NetworkComponents.Layers

7.24 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Layers/TransportLayer.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Layers.TransportLayer
  Transport-Layer

Namensbereiche

- namespace NSA.Model.NetworkComponents.Layers

7.25 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Layerstack.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Layerstack
  Layerstack

Namensbereiche

- namespace NSA.Model.NetworkComponents

7.26 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Network.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Network

Namensbereiche

- namespace NSA.Model.NetworkComponents

7.27 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Route.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Route
  class for a single route of the routingtable

Namensbereiche

- namespace NSA.Model.NetworkComponents

7.28 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/NetworkComponents/Router.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Router

  Implements the network component Router.

Namensbereiche

- namespace NSA.Model.NetworkComponents

7.29 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/NetworkComponents/Switch.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Switch

  Implements the network component switch.

Namensbereiche

- namespace NSA.Model.NetworkComponents

7.30 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/NetworkComponents/Workstation.cs-Dateireferenz

Klassen

- class NSA.Model.NetworkComponents.Workstation

  Implements the network component Workstation.

Namensbereiche

- namespace NSA.Model.NetworkComponents

7.31 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/obj/Debug/TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5A↩ DCB23D92.cs-Dateireferenz

7.32 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/obj/Release/TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5↩ ADCB23D92.cs-Dateireferenz

7.33 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/obj/Debug/TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dda1.cs-Dateireferenz

7.34 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/obj/Release/TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dda1.cs-Dateireferenz

7.35 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/obj/Debug/TemporaryGeneratedFile_E7A71F73-0F8D-4B9B-B56E-8↩ E70B10BC5D3.cs-Dateireferenz

7.36 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/obj/Release/TemporaryGeneratedFile_E7A71F73-0F8D-4B9B-B56E-8↩ E70B10BC5D3.cs-Dateireferenz

7.37 C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/↩ NSA.Model/Properties/AssemblyInfo.cs-Dateireferenz

Index