

NSA.Model

Erzeugt von Doxygen 1.8.11



# Inhaltsverzeichnis

<b>1</b>	<b>Verzeichnis der Namensbereiche</b>	<b>1</b>
1.1	Pakete . . . . .	1
<b>2</b>	<b>Hierarchie-Verzeichnis</b>	<b>3</b>
2.1	Klassenhierarchie . . . . .	3
<b>3</b>	<b>Klassen-Verzeichnis</b>	<b>5</b>
3.1	Auflistung der Klassen . . . . .	5
<b>4</b>	<b>Datei-Verzeichnis</b>	<b>7</b>
4.1	Auflistung der Dateien . . . . .	7
<b>5</b>	<b>Dokumentation der Namensbereiche</b>	<b>9</b>
5.1	NSA-Namensbereichsreferenz . . . . .	9
5.2	NSA.Model-Namensbereichsreferenz . . . . .	9
5.3	NSA.Model.BusinessLogic-Namensbereichsreferenz . . . . .	9
5.4	NSA.Model.NetworkComponents-Namensbereichsreferenz . . . . .	9
5.5	NSA.Model.NetworkComponents.Helper_Classes-Namensbereichsreferenz . . . . .	10
5.6	NSA.Model.NetworkComponents.Layers-Namensbereichsreferenz . . . . .	10

<b>6</b>	<b>Klassen-Dokumentation</b>	<b>11</b>
6.1	NSA.Model.NetworkComponents.Layers.ApplicationLayer Klassenreferenz	11
6.1.1	Dokumentation der Elementfunktionen	12
6.1.1.1	ValidateReceive(Workstation currentNode, ValidationInfo valInfo)	12
6.1.1.2	ValidateSend(Workstation destination, Workstation currentNode, ValidationInfo valInfo)	12
6.2	NSA.Model.NetworkComponents.Connection Klassenreferenz	12
6.2.1	Beschreibung der Konstruktoren und Destruktoren	13
6.2.1.1	Connection(Hardwarenode Source, Hardwarenode Target)	13
6.2.2	Dokumentation der Elementfunktionen	13
6.2.2.1	Equals(object Obj)	13
6.2.2.2	Equals(Connection Other)	14
6.2.2.3	GetHashCode()	14
6.2.2.4	GetPortIndex(Hardwarenode node)	14
6.2.2.5	operator!=(Connection A, Connection B)	15
6.2.2.6	operator==(Connection A, Connection B)	15
6.2.3	Dokumentation der Propertys	15
6.2.3.1	End	15
6.2.3.2	Name	16
6.2.3.3	Start	16
6.3	NSA.Model.NetworkComponents.Layers.DataLinkLayer Klassenreferenz	16
6.3.1	Dokumentation der Elementfunktionen	17
6.3.1.1	ValidateReceive(Workstation currentNode, ValidationInfo valInfo)	17
6.3.1.2	ValidateSend(Workstation destination, Workstation currentNode, ValidationInfo valInfo)	17
6.4	NSA.Model.NetworkComponents.Hardwarenode Klassenreferenz	18
6.4.1	Beschreibung der Konstruktoren und Destruktoren	20
6.4.1.1	Hardwarenode(string Name)	20
6.4.2	Dokumentation der Elementfunktionen	20
6.4.2.1	AddConnection(string IfaceName, Connection Con)	20
6.4.2.2	AddLayer(ILayer Lay)	21

6.4.2.3	Equals(object Obj)	21
6.4.2.4	Equals(Hardwarenode Other)	22
6.4.2.5	GetConnections()	22
6.4.2.6	GetHashCode()	22
6.4.2.7	GetPortIndexOfConnection(Connection c)	23
6.4.2.8	HasIp(IPAddress Ip)	23
6.4.2.9	operator!=(Hardwarenode A, Hardwarenode B)	24
6.4.2.10	operator==(Hardwarenode A, Hardwarenode B)	24
6.4.2.11	Receive(Dictionary< string, object > Tags, ValidationInfo valInfo)	24
6.4.2.12	RemoveConnection(string IfaceName)	24
6.4.2.13	RemoveLayer(ILayer Lay)	25
6.4.2.14	Send(Hardwarenode Destination, Dictionary< string, object > Tags, ValidationInfo valInfo)	25
6.4.3	Dokumentation der Datenelemente	25
6.4.3.1	Connections	25
6.4.3.2	Layerstack	26
6.4.4	Dokumentation der Propertyys	26
6.4.4.1	Name	26
6.5	NSA.Model.NetworkComponents.ILayer Schnittstellenreferenz	26
6.5.1	Dokumentation der Elementfunktionen	27
6.5.1.1	ValidateReceive(Workstation currentNode, ValidationInfo valInfo)	27
6.5.1.2	ValidateSend(Workstation destination, Workstation currentNode, ValidationInfo valInfo)	28
6.6	NSA.Model.NetworkComponents.Interface Klassenreferenz	29
6.6.1	Beschreibung der Konstruktoren und Destruktoren	29
6.6.1.1	Interface(IPAddress Ip, IPAddress Mask, int Number)	29
6.6.2	Dokumentation der Elementfunktionen	29
6.6.2.1	SetInterface(IPAddress Ip, IPAddress Mask)	29
6.6.3	Dokumentation der Propertyys	30
6.6.3.1	IpAddress	30
6.6.3.2	Name	30

6.6.3.3	NamePrefix	30
6.6.3.4	Subnetmask	30
6.7	NSA.Model.NetworkComponents.Helper_Classes.IPAddressExtensions Klassenreferenz	30
6.7.1	Dokumentation der Elementfunktionen	30
6.7.1.1	GetBroadcastAddress(this IPAddress address, IPAddress subnetMask)	30
6.7.1.2	GetNetworkAddress(this IPAddress address, IPAddress subnetMask)	30
6.7.1.3	IsInSameSubnet(this IPAddress address2, IPAddress address, IPAddress subnetMask)	30
6.8	NSA.Model.NetworkComponents.Layerstack Klassenreferenz	31
6.8.1	Beschreibung der Konstruktoren und Destruktoren	31
6.8.1.1	Layerstack()	31
6.8.2	Dokumentation der Elementfunktionen	31
6.8.2.1	AddLayer(ILayer lay)	31
6.8.2.2	GetLayer(int index)	32
6.8.2.3	GetSize()	32
6.8.2.4	RemoveLayer(ILayer lay)	32
6.9	NSA.Model.NetworkComponents.Network Klassenreferenz	33
6.9.1	Beschreibung der Konstruktoren und Destruktoren	34
6.9.1.1	Network()	34
6.9.2	Dokumentation der Elementfunktionen	34
6.9.2.1	AddConnection(string StartNodeInterfaceName, string EndNodeInterfaceName, Connection newConnection)	34
6.9.2.2	AddHardwarenode(Hardwarenode newNode)	34
6.9.2.3	GetAllHardwarenodes()	34
6.9.2.4	GetAllWorkstations()	35
6.9.2.5	GetConnectionByName(string Name)	35
6.9.2.6	GetHardwarenodeByName(string Name)	35
6.9.2.7	GetRouters()	35
6.9.2.8	GetWorkstationByIP(IPAddress ip)	35
6.9.2.9	RemoveConnection(string ConnectionName)	36
6.9.2.10	RemoveHardwarnode(string name)	36

6.10 NSA.Model.NetworkComponents.Layers.NetworkLayer Klassenreferenz . . . . .	37
6.10.1 Dokumentation der Elementfunktionen . . . . .	37
6.10.1.1 ValidateReceive(Workstation currentNode, ValidationInfo valInfo) . . . . .	37
6.10.1.2 ValidateSend(Workstation destination, Workstation currentNode, ValidationInfo valInfo) . . . . .	38
6.11 NSA.Model.BusinessLogic.NodeLocation Klassenreferenz . . . . .	38
6.11.1 Dokumentation der Propertyys . . . . .	39
6.11.1.1 Name . . . . .	39
6.11.1.2 Point . . . . .	39
6.12 NSA.Model.BusinessLogic.Packet Klassenreferenz . . . . .	39
6.12.1 Beschreibung der Konstruktoren und Destruktoren . . . . .	39
6.12.1.1 Packet(Hardwarenode _source, Hardwarenode _destination, int _ttl, Dictionary< string, object > _tags, bool expRes) . . . . .	39
6.12.2 Dokumentation der Elementfunktionen . . . . .	39
6.12.2.1 Send() . . . . .	39
6.12.3 Dokumentation der Propertyys . . . . .	40
6.12.3.1 expectedResult . . . . .	40
6.12.3.2 result . . . . .	40
6.13 NSA.Model.NetworkComponents.Layers.PhysicalLayer Klassenreferenz . . . . .	40
6.13.1 Dokumentation der Elementfunktionen . . . . .	41
6.13.1.1 ValidateReceive(Workstation currentNode, ValidationInfo valInfo) . . . . .	41
6.13.1.2 ValidateSend(Workstation destination, Workstation currentNode, ValidationInfo valInfo) . . . . .	41
6.14 NSA.Model.NetworkComponents.Layers.PresentationLayer Klassenreferenz . . . . .	42
6.14.1 Dokumentation der Elementfunktionen . . . . .	42
6.14.1.1 ValidateReceive(Workstation currentNode, ValidationInfo valInfo) . . . . .	42
6.14.1.2 ValidateSend(Workstation destination, Workstation currentNode, ValidationInfo valInfo) . . . . .	43
6.15 NSA.Model.BusinessLogic.Project Klassenreferenz . . . . .	43
6.15.1 Beschreibung der Konstruktoren und Destruktoren . . . . .	43
6.15.1.1 Project() . . . . .	43
6.15.2 Dokumentation der Propertyys . . . . .	44

6.15.2.1	Network	44
6.15.2.2	NodeLocations	44
6.15.2.3	Path	44
6.16	NSA.Model.NetworkComponents.Helper_Classes.Result Klassenreferenz	44
6.16.1	Beschreibung der Konstruktoren und Destruktoren	44
6.16.1.1	Result()	44
6.16.2	Dokumentation der Property's	44
6.16.2.1	ErrorID	44
6.16.2.2	LayerError	44
6.16.2.3	Res	44
6.16.2.4	SendError	44
6.17	NSA.Model.NetworkComponents.Route Klassenreferenz	45
6.17.1	Beschreibung der Konstruktoren und Destruktoren	45
6.17.1.1	Route(IPAddress Destination, IPAddress Subnetmask, IPAddress Gateway, Interface Iface)	45
6.17.2	Dokumentation der Elementfunktionen	45
6.17.2.1	SetRoute(IPAddress DestinationIp, IPAddress Mask, IPAddress Gateway↔ Address, Interface Interface)	45
6.17.3	Dokumentation der Property's	46
6.17.3.1	Destination	46
6.17.3.2	Gateway	46
6.17.3.3	Iface	46
6.17.3.4	Name	46
6.17.3.5	Subnetmask	46
6.18	NSA.Model.NetworkComponents.Router Klassenreferenz	46
6.18.1	Beschreibung der Konstruktoren und Destruktoren	47
6.18.1.1	Router(string Name)	47
6.18.2	Dokumentation der Property's	48
6.18.2.1	IsGateway	48
6.19	NSA.Model.BusinessLogic.Rule Klassenreferenz	48
6.19.1	Beschreibung der Konstruktoren und Destruktoren	48



6.19.1.1	Rule()	48
6.19.2	Dokumentation der Propertys	48
6.19.2.1	Applicable	48
6.19.2.2	HasInternet	48
6.19.2.3	Name	48
6.19.2.4	NodeNames	48
6.19.2.5	OnlyNodeNames	48
6.19.2.6	Ssl	48
6.19.2.7	SubnetNames	48
6.19.2.8	Ttl	48
6.20	NSA.Model.NetworkComponents.Layers.SessionLayer Klassenreferenz	49
6.20.1	Dokumentation der Elementfunktionen	49
6.20.1.1	ValidateReceive(Workstation currentNode, ValidationInfo valInfo)	49
6.20.1.2	ValidateSend(Workstation destination, Workstation currentNode, ValidationInfo valInfo)	50
6.21	NSA.Model.BusinessLogic.Simulation Klassenreferenz	50
6.21.1	Beschreibung der Konstruktoren und Destruktoren	50
6.21.1.1	Simulation(int _id)	50
6.21.2	Dokumentation der Elementfunktionen	50
6.21.2.1	AddPacketSend(Packet packet)	50
6.21.2.2	Execute()	51
6.22	NSA.Model.NetworkComponents.Switch Klassenreferenz	51
6.22.1	Beschreibung der Konstruktoren und Destruktoren	52
6.22.1.1	Switch(string Name)	52
6.22.2	Dokumentation der Elementfunktionen	53
6.22.2.1	AddInterface()	53
6.22.2.2	GetInterfaceCount()	53
6.22.2.3	RemoveInterface(string InterfaceName)	53
6.22.2.4	Send(Hardwarenode Destination, Dictionary< string, object > Tags, ValidationInfo valInfo)	53
6.22.2.5	SendTolp(ValidationInfo valInfo)	54

6.22.3	Dokumentation der Propertyts	54
6.22.3.1	Interfaces	54
6.23	NSA.Model.BusinessLogic.Testscenario Klassenreferenz	55
6.23.1	Beschreibung der Konstruktoren und Destruktoren	55
6.23.1.1	Testscenario()	55
6.23.2	Dokumentation der Elementfunktionen	55
6.23.2.1	ParseRulesToNetwork()	55
6.23.3	Dokumentation der Propertyts	55
6.23.3.1	Id	55
6.23.3.2	Rules	55
6.24	NSA.Model.NetworkComponents.Layers.TransportLayer Klassenreferenz	55
6.24.1	Dokumentation der Elementfunktionen	56
6.24.1.1	ValidateReceive(Workstation currentNode, ValidationInfo valInfo)	56
6.24.1.2	ValidateSend(Workstation destination, Workstation currentNode, ValidationInfo valInfo)	56
6.25	NSA.Model.NetworkComponents.Helper_Classes.ValidationInfo Klassenreferenz	57
6.25.1	Dokumentation der Propertyts	57
6.25.1.1	Iface	57
6.25.1.2	NextNodeIP	57
6.25.1.3	NextNodes	57
6.25.1.4	Res	57
6.26	NSA.Model.NetworkComponents.Workstation Klassenreferenz	58
6.26.1	Beschreibung der Konstruktoren und Destruktoren	59
6.26.1.1	Workstation(string Name)	59
6.26.2	Dokumentation der Elementfunktionen	60
6.26.2.1	AddInterface(IPAddress Ip, IPAddress Subnetmask)	60
6.26.2.2	AddRoute(Route Route)	60
6.26.2.3	GetInterfaceCount()	60
6.26.2.4	GetInterfaces()	60
6.26.2.5	GetRouteAt(int Index)	61
6.26.2.6	GetRouteCount()	61
6.26.2.7	GetRoutes()	61
6.26.2.8	HasIp(IPAddress Ip)	61
6.26.2.9	Receive(Dictionary< string, object > Tags, ValidationInfo valInfo)	62
6.26.2.10	RemoveInterface(string InterfaceName)	62
6.26.2.11	RemoveRoute(string N)	63
6.26.2.12	Send(Hardwarenode Destination, Dictionary< string, object > Tags, ValidationInfo valInfo)	63
6.26.2.13	SetInterface(string Ifacename, IPAddress Ip, IPAddress Mask)	63
6.26.2.14	SetRoute(string RouteName, IPAddress Destination, IPAddress Subnetmask, IPAddress Gateway, Interface Iface)	64
6.26.3	Dokumentation der Propertyts	64
6.26.3.1	StandardGateway	64
6.26.3.2	StandardGatewayPort	64

<b>7</b>	<b>Datei-Dokumentation</b>	<b>65</b>
7.1	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ BusinessLogic/Packet.cs-Dateireferenz . . . . .	65
7.2	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ BusinessLogic/Project.cs-Dateireferenz . . . . .	65
7.3	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ BusinessLogic/Simulation.cs-Dateireferenz . . . . .	65
7.4	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ BusinessLogic/Testscenario.cs-Dateireferenz . . . . .	66
7.5	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Connection.cs-Dateireferenz . . . . .	66
7.6	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Hardwarenode.cs-Dateireferenz . . . . .	66
7.7	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Helper Classes/IPAddressExtensions.cs-Dateireferenz . . . . .	66
7.8	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Helper Classes/Result.cs-Dateireferenz . . . . .	67
7.9	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Helper Classes/ValidationInfo.cs-Dateireferenz . . . . .	67
7.10	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/ILayer.cs-Dateireferenz . . . . .	67
7.11	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Interface.cs-Dateireferenz . . . . .	67
7.12	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Layers/ApplicationLayer.cs-Dateireferenz . . . . .	68
7.13	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Layers/DataLinkLayer.cs-Dateireferenz . . . . .	68
7.14	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Layers/NetworkLayer.cs-Dateireferenz . . . . .	68
7.15	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Layers/PhysicalLayer.cs-Dateireferenz . . . . .	68
7.16	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Layers/PresentationLayer.cs-Dateireferenz . . . . .	69
7.17	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Layers/SessionLayer.cs-Dateireferenz . . . . .	69
7.18	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Layers/TransportLayer.cs-Dateireferenz . . . . .	69
7.19	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Layerstack.cs-Dateireferenz . . . . .	69

7.20	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Network.cs-Dateireferenz . . . . .	70
7.21	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Route.cs-Dateireferenz . . . . .	70
7.22	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Router.cs-Dateireferenz . . . . .	70
7.23	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Switch.cs-Dateireferenz . . . . .	70
7.24	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Workstation.cs-Dateireferenz . . . . .	71
7.25	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Debug/TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs-Dateireferenz . . . . .	71
7.26	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Debug/TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dda1.cs-Dateireferenz . . . . .	71
7.27	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Debug/TemporaryGeneratedFile_E7A71F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs-Dateireferenz . . . . .	71
7.28	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ Properties/AssemblyInfo.cs-Dateireferenz . . . . .	71
<b>Index</b>		<b>73</b>

# Kapitel 1

## Verzeichnis der Namensbereiche

### 1.1 Pakete

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

<a href="#">NSA</a> . . . . .	9
<a href="#">NSA.Model</a> . . . . .	9
<a href="#">NSA.Model.BusinessLogic</a> . . . . .	9
<a href="#">NSA.Model.NetworkComponents</a> . . . . .	9
<a href="#">NSA.Model.NetworkComponents.Helper_Classes</a> . . . . .	10
<a href="#">NSA.Model.NetworkComponents.Layers</a> . . . . .	10



## Kapitel 2

# Hierarchie-Verzeichnis

### 2.1 Klassenhierarchie

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

NSA.Model.NetworkComponents.Connection . . . . .	12
NSA.Model.NetworkComponents.Hardwarenode . . . . .	18
NSA.Model.NetworkComponents.Switch . . . . .	51
NSA.Model.NetworkComponents.Workstation . . . . .	58
NSA.Model.NetworkComponents.Router . . . . .	46
NSA.Model.NetworkComponents.ILayer . . . . .	26
NSA.Model.NetworkComponents.Layers.ApplicationLayer . . . . .	11
NSA.Model.NetworkComponents.Layers.DataLinkLayer . . . . .	16
NSA.Model.NetworkComponents.Layers.NetworkLayer . . . . .	37
NSA.Model.NetworkComponents.Layers.PhysicalLayer . . . . .	40
NSA.Model.NetworkComponents.Layers.PresentationLayer . . . . .	42
NSA.Model.NetworkComponents.Layers.SessionLayer . . . . .	49
NSA.Model.NetworkComponents.Layers.TransportLayer . . . . .	55
NSA.Model.NetworkComponents.Interface . . . . .	29
NSA.Model.NetworkComponents.Helper_Classes.IPAddressExtensions . . . . .	30
NSA.Model.NetworkComponents.Layerstack . . . . .	31
NSA.Model.NetworkComponents.Network . . . . .	33
NSA.Model.BusinessLogic.NodeLocation . . . . .	38
NSA.Model.BusinessLogic.Packet . . . . .	39
NSA.Model.BusinessLogic.Project . . . . .	43
NSA.Model.NetworkComponents.Helper_Classes.Result . . . . .	44
NSA.Model.NetworkComponents.Route . . . . .	45
NSA.Model.BusinessLogic.Rule . . . . .	48
NSA.Model.BusinessLogic.Simulation . . . . .	50
NSA.Model.BusinessLogic.Testscenario . . . . .	55
NSA.Model.NetworkComponents.Helper_Classes.ValidationInfo . . . . .	57





## Kapitel 3

# Klassen-Verzeichnis

### 3.1 Auflistung der Klassen

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

NSA.Model.NetworkComponents.Layers.ApplicationLayer	11
NSA.Model.NetworkComponents.Connection	12
NSA.Model.NetworkComponents.Layers.DataLinkLayer	16
NSA.Model.NetworkComponents.HardwareNode	18
NSA.Model.NetworkComponents.ILayer	26
NSA.Model.NetworkComponents.Interface	29
NSA.Model.NetworkComponents.Helper_Classes.IPAddressExtensions	30
NSA.Model.NetworkComponents.Layerstack	31
NSA.Model.NetworkComponents.Network	33
NSA.Model.NetworkComponents.Layers.NetworkLayer	37
NSA.Model.BusinessLogic.NodeLocation	38
NSA.Model.BusinessLogic.Packet	39
NSA.Model.NetworkComponents.Layers.PhysicalLayer	40
NSA.Model.NetworkComponents.Layers.PresentationLayer	42
NSA.Model.BusinessLogic.Project	43
NSA.Model.NetworkComponents.Helper_Classes.Result	44
NSA.Model.NetworkComponents.Route	45
NSA.Model.NetworkComponents.Router	46
NSA.Model.BusinessLogic.Rule	48
NSA.Model.NetworkComponents.Layers.SessionLayer	49
NSA.Model.BusinessLogic.Simulation	50
NSA.Model.NetworkComponents.Switch	51
NSA.Model.BusinessLogic.Testscenario	55
NSA.Model.NetworkComponents.Layers.TransportLayer	55
NSA.Model.NetworkComponents.Helper_Classes.ValidationInfo	57
NSA.Model.NetworkComponents.Workstation	58



## Kapitel 4

# Datei-Verzeichnis

### 4.1 Auflistung der Dateien

Hier folgt die Aufzählung aller Dateien mit einer Kurzbeschreibung:

C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↔ Logic/ <a href="#">Packet.cs</a> . . . . .	65
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↔ Logic/ <a href="#">Project.cs</a> . . . . .	65
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↔ Logic/ <a href="#">Simulation.cs</a> . . . . .	65
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↔ Logic/ <a href="#">Testscenario.cs</a> . . . . .	66
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Connection.cs</a> . . . . .	66
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Hardwarenode.cs</a> . . . . .	66
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">ILayer.cs</a> . . . . .	67
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Interface.cs</a> . . . . .	67
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Layerstack.cs</a> . . . . .	69
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Network.cs</a> . . . . .	70
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Route.cs</a> . . . . .	70
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Router.cs</a> . . . . .	70
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Switch.cs</a> . . . . .	70
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Workstation.cs</a> . . . . .	71
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Helper Classes/ <a href="#">IPAddressExtensions.cs</a> . . . . .	66
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Helper Classes/ <a href="#">Result.cs</a> . . . . .	67
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Helper Classes/ <a href="#">ValidationInfo.cs</a> . . . . .	67
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Layers/ <a href="#">ApplicationLayer.cs</a> . . . . .	68

C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Layers/ <a href="#">DataLinkLayer.cs</a> . . . . .	68
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Layers/ <a href="#">NetworkLayer.cs</a> . . . . .	68
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Layers/ <a href="#">PhysicalLayer.cs</a> . . . . .	68
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Layers/ <a href="#">PresentationLayer.cs</a> . . . . .	69
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Layers/ <a href="#">SessionLayer.cs</a> . . . . .	69
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Layers/ <a href="#">TransportLayer.cs</a> . . . . .	69
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Debug/ <a href="#">TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs</a> . . . . .	71
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Debug/ <a href="#">TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dda1.cs</a> . . . . .	71
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Debug/ <a href="#">TemporaryGeneratedFile_E7A71F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs</a> . . . . .	71
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Properties/ <a href="#">Assembly↔ Info.cs</a> . . . . .	71

## 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

#### Klassen

- class [NodeLocation](#)
- class [Packet](#)
- class [Project](#)
- class [Rule](#)
- class [Simulation](#)
- class [Testscenario](#)

### 5.4 NSA.Model.NetworkComponents-Namensbereichsreferenz

#### Namensbereiche

- namespace [Helper\\_Classes](#)
- namespace [Layers](#)

## Klassen

- class [Connection](#)
- class [Hardwarenode](#)
- interface [ILayer](#)
- class [Interface](#)
- class [Layerstack](#)
- class [Network](#)
- class [Route](#)
- class [Router](#)
- class [Switch](#)
- class [Workstation](#)

## 5.5 NSA.Model.NetworkComponents.Helper\_Classes-Namensbereichsreferenz

### Klassen

- class [IPAddressExtensions](#)
- class [Result](#)
- class [ValidationInfo](#)

## 5.6 NSA.Model.NetworkComponents.Layers-Namensbereichsreferenz

### Klassen

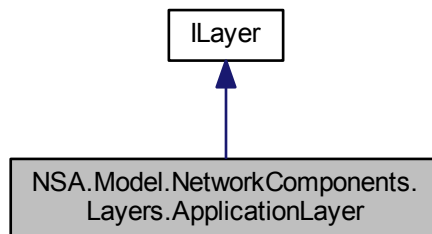
- class [ApplicationLayer](#)
- class [DataLinkLayer](#)
- class [NetworkLayer](#)
- class [PhysicalLayer](#)
- class [PresentationLayer](#)
- class [SessionLayer](#)
- class [TransportLayer](#)

## Kapitel 6

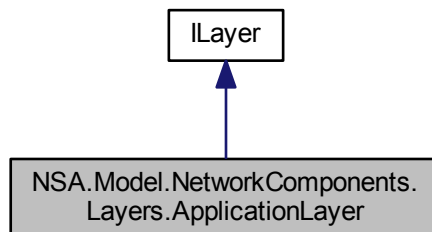
# Klassen-Dokumentation

### 6.1 NSA.Model.NetworkComponents.Layers.ApplicationLayer Klassenreferenz

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



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



## Öffentliche Methoden

- bool [ValidateReceive](#) ([Workstation](#) currentNode, [ValidationInfo](#) valInfo)  
*Validates the layer while receiving a packet.*
- void [ValidateSend](#) ([Workstation](#) destination, [Workstation](#) currentNode, [ValidationInfo](#) valInfo)  
*Validates the layer while sending a packet.*

### 6.1.1 Dokumentation der Elementfunktionen

6.1.1.1 bool [NSA.Model.NetworkComponents.Layers.ApplicationLayer.ValidateReceive](#) ( [Workstation](#) *currentNode*, [ValidationInfo](#) *valInfo* )

Validates the layer while receiving a packet.

#### Parameter

<i>currentNode</i>	
<i>valInfo</i>	

#### Rückgabe

Boolean value indicating if the validation was successfull

Implementiert [NSA.Model.NetworkComponents.ILayer](#).

6.1.1.2 void [NSA.Model.NetworkComponents.Layers.ApplicationLayer.ValidateSend](#) ( [Workstation](#) *destination*, [Workstation](#) *currentNode*, [ValidationInfo](#) *valInfo* )

Validates the layer while sending a packet.

#### Parameter

<i>destination</i>	The destination.
<i>currentNode</i>	The current node.
<i>valInfo</i>	

Implementiert [NSA.Model.NetworkComponents.ILayer](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Layers/[ApplicationLayer.cs](#)

## 6.2 NSA.Model.NetworkComponents.Connection Klassenreferenz

### 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.*
- 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 !=.*

## Geschützte Methoden

- bool [Equals](#) ([Connection](#) Other)  
*Equalses the specified other.*

## Property

- [Hardwarenode Start](#) [get]
- [Hardwarenode End](#) [get]
- string [Name](#) [get]

### 6.2.1 Beschreibung der Konstruktoren und Destruktoren

#### 6.2.1.1 NSA.Model.NetworkComponents.Connection.Connection ( [Hardwarenode Source](#), [Hardwarenode Target](#) )

Initializes a new instance of the [Connection](#) class.

Parameter

<i>Source</i>	The sourcenode.
<i>Target</i>	The targetnode.

### 6.2.2 Dokumentation der Elementfunktionen

#### 6.2.2.1 override bool NSA.Model.NetworkComponents.Connection.Equals ( object Obj )

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

## Parameter

<i>Obj</i>	The System.Object to compare with this instance.
------------	--

## Rückgabe

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

**6.2.2.2 bool NSA.Model.NetworkComponents.Connection.Equals ( Connection *Other* ) [protected]**

Equalses the specified other.

## Parameter

<i>Other</i>	The other.
--------------	------------

## Rückgabe

**6.2.2.3 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.

**6.2.2.4 int NSA.Model.NetworkComponents.Connection.GetPortIndex ( Hardwarenode *node* )**

Gets the index of the port.

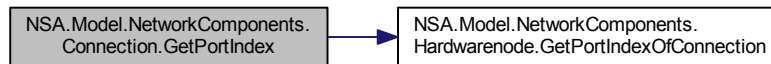
## Parameter

<i>node</i>	The node.
-------------	-----------

**Rückgabe**

Portindex

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



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

Implements the operator !=.

**Parameter**

<i>A</i>	a.
<i>B</i>	The b.

**Rückgabe**

The result of the operator.

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

Implements the operator ==.

**Parameter**

<i>A</i>	a.
<i>B</i>	The b.

**Rückgabe**

The result of the operator.

**6.2.3 Dokumentation der Property**

**6.2.3.1** `Hardwarenode NSA.Model.NetworkComponents.Connection.End` `[get]`

6.2.3.2 `string NSA.Model.NetworkComponents.Connection.Name` [get]

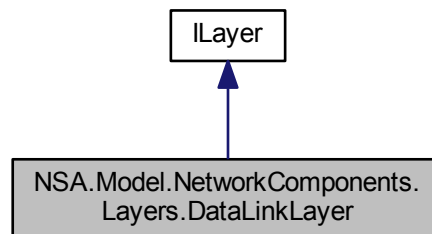
6.2.3.3 `Hardwarenode NSA.Model.NetworkComponents.Connection.Start` [get]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

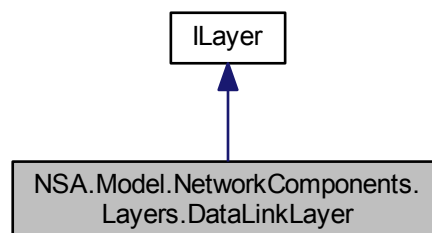
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/[Connection.cs](#)

## 6.3 NSA.Model.NetworkComponents.Layers.DataLinkLayer Klassenreferenz

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



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



### Öffentliche Methoden

- void `ValidateSend` (`Workstation` destination, `Workstation` currentNode, `ValidationInfo` valInfo)  
*Validates the layer while sending a packet.*
- bool `ValidateReceive` (`Workstation` currentNode, `ValidationInfo` valInfo)  
*Validates the layer while receiving a packet.*

### 6.3.1 Dokumentation der Elementfunktionen

#### 6.3.1.1 `bool NSA.Model.NetworkComponents.Layers.DataLinkLayer.ValidateReceive ( Workstation currentNode, ValidationInfo valInfo )`

Validates the layer while receiving a packet.

##### Parameter

<i>currentNode</i>	
<i>valInfo</i>	

##### Rückgabe

Boolean value indicating if the validation was successful

Implementiert [NSA.Model.NetworkComponents.ILayer](#).

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



#### 6.3.1.2 `void NSA.Model.NetworkComponents.Layers.DataLinkLayer.ValidateSend ( Workstation destination, Workstation currentNode, ValidationInfo valInfo )`

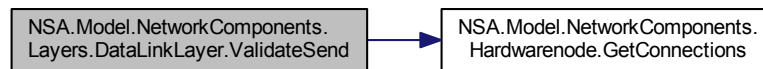
Validates the layer while sending a packet.

##### Parameter

<i>destination</i>	The destination.
<i>currentNode</i>	The current node.
<i>valInfo</i>	

Implementiert [NSA.Model.NetworkComponents.ILayer](#).

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

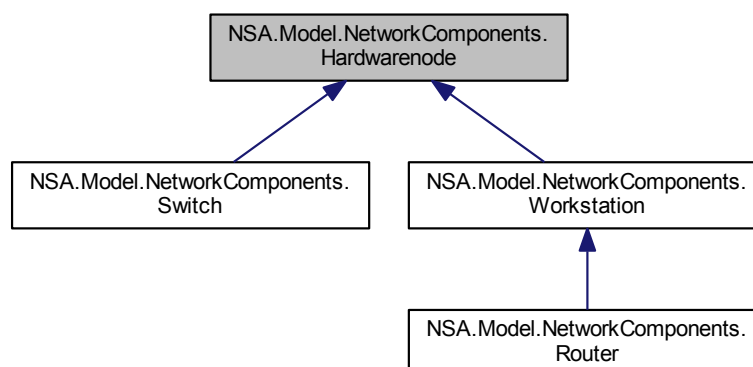


Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

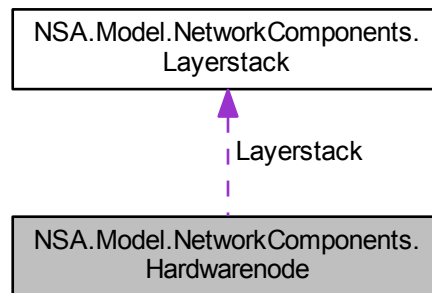
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Layers/[DataLinkLayer.cs](#)

## 6.4 NSA.Model.NetworkComponents.Hardwarenode Klassenreferenz

Klassendiagramm für NSA.Model.NetworkComponents.Hardwarenode:



Zusammengehörigkeiten von NSA.Model.NetworkComponents.Hardwarenode:



## Öffentliche Methoden

- [Hardwarenode](#) (string [Name](#))  
*Initializes a new instance of the [Hardwarenode](#) class.*
- void [AddConnection](#) (string [IfaceName](#), [Connection](#) [Con](#))  
*Adds a connection.*
- virtual void [RemoveConnection](#) (string [IfaceName](#))  
*Removes a connection.*
- void [AddLayer](#) ([ILayer](#) [Lay](#))  
*Adds a layer to the layerstack.*
- void [RemoveLayer](#) ([ILayer](#) [Lay](#))  
*Removes a layer from the layerstack.*
- virtual bool [HasIp](#) ([IPAddress](#) [Ip](#))  
*Checks if the [Hardwarenode](#) has the IP*
- virtual List< [Hardwarenode](#) > [Send](#) ([Hardwarenode](#) [Destination](#), Dictionary< string, object > [Tags](#), [ValidationInfo](#) [vallInfo](#))  
*[Hardwarenode](#) sends the package to specified destination.*
- virtual bool [Receive](#) (Dictionary< string, object > [Tags](#), [ValidationInfo](#) [vallInfo](#))  
*[Hardwarenode](#) receives the package.*
- Dictionary< string, [Connection](#) > [GetConnections](#) ()  
*Gets the connections.*
- override bool [Equals](#) (object [Obj](#))  
*Determines whether the specified System.Object, is equal to this instance.*
- 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

- bool [Equals](#) ([Hardwarenode](#) Other)  
*Equalses the specified other.*

## Geschützte Attribute

- [Layerstack](#) [Layerstack](#) = new [Layerstack](#)()
- Dictionary< string, [Connection](#) > [Connections](#) = new Dictionary<string, [Connection](#)>()

## Propertys

- string [Name](#) [get, set]

### 6.4.1 Beschreibung der Konstruktoren und Destruktoren

#### 6.4.1.1 NSA.Model.NetworkComponents.Hardwarenode.Hardwarenode ( string Name )

Initializes a new instance of the [Hardwarenode](#) class.

##### Parameter

<i>Name</i>	The name.
-------------	-----------

### 6.4.2 Dokumentation der Elementfunktionen

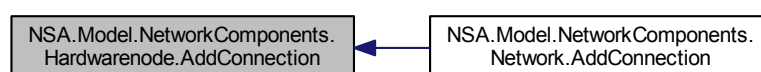
#### 6.4.2.1 void NSA.Model.NetworkComponents.Hardwarenode.AddConnection ( string IfaceName, Connection Con )

Adds a connection.

##### Parameter

<i>IfaceName</i>	Name of the interface where the connection should be added.
<i>Con</i>	The connection to be added.

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





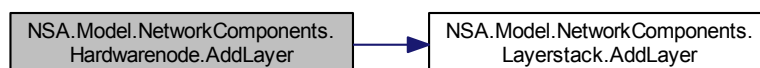
#### 6.4.2.2 void NSA.Model.NetworkComponents.Hardwarenode.AddLayer ( ILayer Lay )

Adds a layer to the layerstack.

##### Parameter

<i>Lay</i>	The layer to be added.
------------	------------------------

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



#### 6.4.2.3 override bool NSA.Model.NetworkComponents.Hardwarenode.Equals ( object Obj )

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

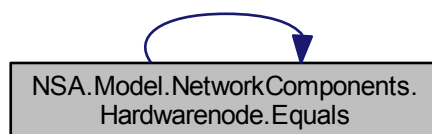
##### Parameter

<i>Obj</i>	The System.Object to compare with this instance.
------------	--

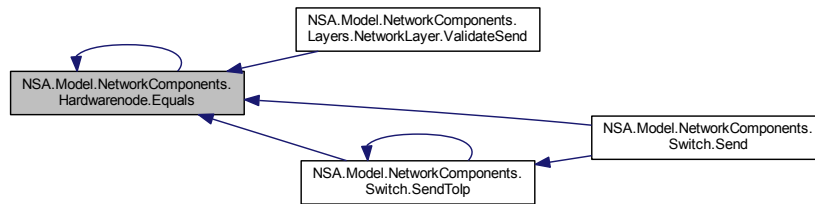
##### Rückgabe

`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:



#### 6.4.2.4 `bool NSA.Model.NetworkComponents.Hardwarenode.Equals ( Hardwarenode Other ) [protected]`

Equalses the specified other.

Parameter

<i>Other</i>	The other.
--------------	------------

Rückgabe

#### 6.4.2.5 `Dictionary<string, Connection> NSA.Model.NetworkComponents.Hardwarenode.GetConnections ( )`

Gets the connections.

Rückgabe

Connections

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



#### 6.4.2.6 `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.

#### 6.4.2.7 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:



#### 6.4.2.8 virtual bool NSA.Model.NetworkComponents.Hardwarenode.HasIp ( IPAddress Ip ) [virtual]

Checks if the [Hardwarenode](#) has the IP

##### Parameter

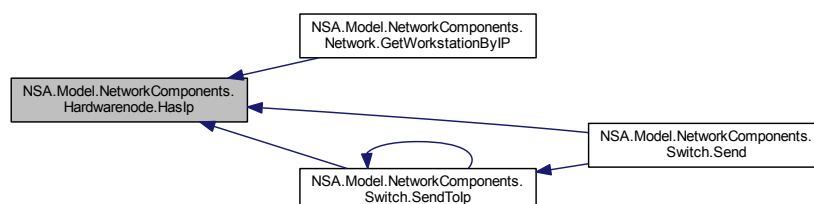
Ip	The ip.
----	---------

##### Rückgabe

A bool

Erneute Implementation in [NSA.Model.NetworkComponents.Workstation](#).

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



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

Implements the operator !=.

Parameter

<i>A</i>	a.
<i>B</i>	The b.

Rückgabe

The result of the operator.

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

Implements the operator ==.

Parameter

<i>A</i>	a.
<i>B</i>	The b.

Rückgabe

The result of the operator.

6.4.2.11 `virtual bool NSA.Model.NetworkComponents.Hardwarenode.Receive ( Dictionary< string, object > Tags, ValidationInfo valInfo )` `[virtual]`

[Hardwarenode](#) receives the package.

Parameter

<i>Tags</i>	Optional tags.
<i>valInfo</i>	The validation Info

Rückgabe

If the [Hardwarenode](#) could receive the package

Erneute Implementation in [NSA.Model.NetworkComponents.Workstation](#).

6.4.2.12 `virtual void NSA.Model.NetworkComponents.Hardwarenode.RemoveConnection ( string IfaceName )` `[virtual]`

Removes a connection.

## Parameter

<i>InterfaceName</i>	Name of the interface where the connection should be removed.
----------------------	---

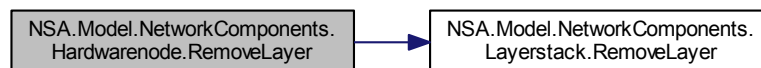
## 6.4.2.13 void NSA.Model.NetworkComponents.Hardwarenode.RemoveLayer ( ILayer Lay )

Removes a layer from the layerstack.

## Parameter

<i>Lay</i>	The layer to be removed.
------------	--------------------------

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



## 6.4.2.14 virtual List&lt;Hardwarenode&gt; NSA.Model.NetworkComponents.Hardwarenode.Send ( Hardwarenode Destination, Dictionary&lt; string, object &gt; Tags, ValidationInfo valInfo ) [virtual]

[Hardwarenode](#) sends the package to specified destination.

## Parameter

<i>Destination</i>	The destination.
<i>Tags</i>	Optional tags.
<i>valInfo</i>	

## Rückgabe

The [Hardwarenode](#) which received the package or null if an error occurred

Erneute Implementation in [NSA.Model.NetworkComponents.Workstation](#) und [NSA.Model.NetworkComponents.Switch](#).

## 6.4.3 Dokumentation der Datenelemente

## 6.4.3.1 Dictionary&lt;string, Connection&gt; NSA.Model.NetworkComponents.Hardwarenode.Connections = new Dictionary&lt;string, Connection&gt;() [protected]

6.4.3.2 `Layerstack NSA.Model.NetworkComponents.Hardwarenode.Layerstack = new Layerstack()` [protected]

## 6.4.4 Dokumentation der Propertys

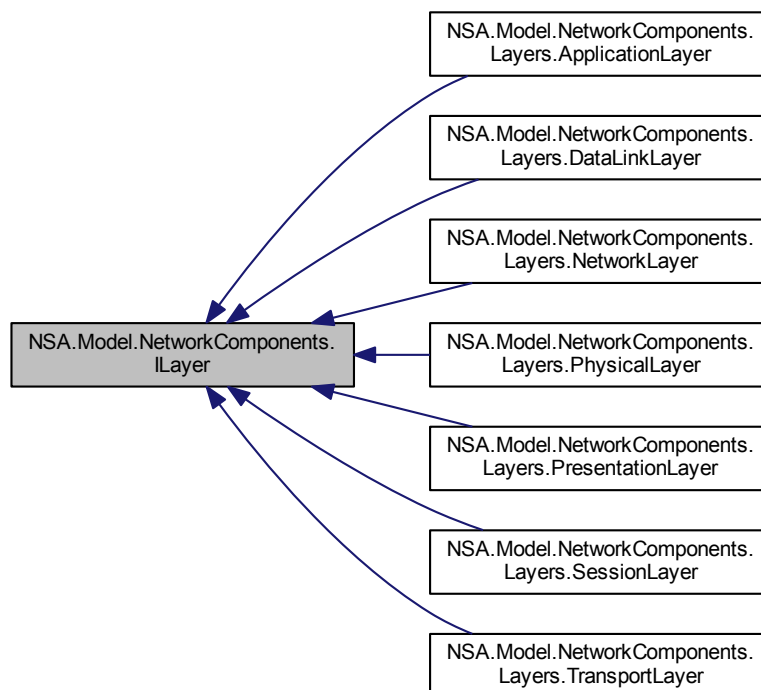
6.4.4.1 `string NSA.Model.NetworkComponents.Hardwarenode.Name` [get], [set]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/[Hardwarenode.cs](#)

## 6.5 NSA.Model.NetworkComponents.ILayer Schnittstellenreferenz

Klassendiagramm für NSA.Model.NetworkComponents.ILayer:



## Öffentliche Methoden

- void `ValidateSend` (`Workstation` destination, `Workstation` currentNode, `ValidationInfo` valInfo)  
*Validates the layer while sending a packet.*
- bool `ValidateReceive` (`Workstation` currentNode, `ValidationInfo` valInfo)  
*Validates the layer while receiving a packet.*

### 6.5.1 Dokumentation der Elementfunktionen

6.5.1.1 `bool NSA.Model.NetworkComponents.ILayer.ValidateReceive ( Workstation currentNode, ValidationInfo valInfo )`

Validates the layer while receiving a packet.

## Parameter

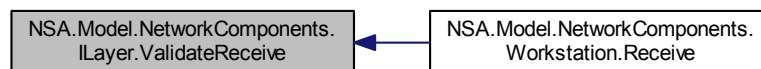
<i>currentNode</i>	
<i>valInfo</i>	

## Rückgabe

Boolean value indicating if the validation was successfull

Implementiert in [NSA.Model.NetworkComponents.Layers.DataLinkLayer](#), [NSA.Model.NetworkComponents.Layers.ApplicationLayer](#), [NSA.Model.NetworkComponents.Layers.NetworkLayer](#), [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:



6.5.1.2 void NSA.Model.NetworkComponents.ILayer.ValidateSend ( Workstation *destination*, Workstation *currentNode*, ValidationInfo *valInfo* )

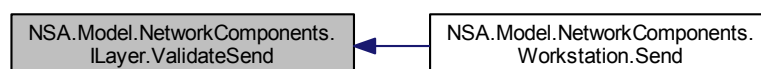
Validates the layer while sending a packet.

## Parameter

<i>destination</i>	The destination.
<i>currentNode</i>	The current node.
<i>valInfo</i>	

Implementiert in [NSA.Model.NetworkComponents.Layers.ApplicationLayer](#), [NSA.Model.NetworkComponents.Layers.NetworkLayer](#), [NSA.Model.NetworkComponents.Layers.PhysicalLayer](#), [NSA.Model.NetworkComponents.Layers.PresentationLayer](#), [NSA.Model.NetworkComponents.Layers.SessionLayer](#), [NSA.Model.NetworkComponents.Layers.TransportLayer](#) und [NSA.Model.NetworkComponents.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/NetworkComponents/ILayer.cs

## 6.6 NSA.Model.NetworkComponents.Interface Klassenreferenz

### 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.*

### Property

- string [Name](#) [get]
- IPAddress [IpAddress](#) [get, set]
- IPAddress [Subnetmask](#) [get, set]
- static string [NamePrefix](#) [get]

### 6.6.1 Beschreibung der Konstruktoren und Destruktoren

#### 6.6.1.1 NSA.Model.NetworkComponents.Interface.Interface ( IPAddress Ip, IPAddress Mask, int Number )

Initializes a new instance of the [Interface](#) class.

##### Parameter

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

### 6.6.2 Dokumentation der Elementfunktionen

#### 6.6.2.1 void NSA.Model.NetworkComponents.Interface.SetInterface ( IPAddress Ip, IPAddress Mask )

Sets the interface.

##### Parameter

<i>Ip</i>	The new ip.
<i>Mask</i>	The new subnetmask.

### 6.6.3 Dokumentation der Propertys

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

6.6.3.2 `string NSA.Model.NetworkComponents.Interface.Name` `[get]`

6.6.3.3 `string NSA.Model.NetworkComponents.Interface.NamePrefix` `[static], [get]`

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

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/[Interface.cs](#)

## 6.7 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)

### 6.7.1 Dokumentation der Elementfunktionen

6.7.1.1 `static IPAddress NSA.Model.NetworkComponents.Helper_Classes.IpAddressExtensions.GetBroadcastAddress ( this IPAddress address, IPAddress subnetMask )` `[static]`

6.7.1.2 `static IPAddress NSA.Model.NetworkComponents.Helper_Classes.IpAddressExtensions.GetNetworkAddress ( this IPAddress address, IPAddress subnetMask )` `[static]`

6.7.1.3 `static bool NSA.Model.NetworkComponents.Helper_Classes.IpAddressExtensions.IsInSameSubnet ( this IPAddress address2, IPAddress address, 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](#)

## 6.8 NSA.Model.NetworkComponents.Layerstack Klassenreferenz

### Öffentliche Methoden

- [Layerstack](#) ()  
*Initializes a new instance of the [Layerstack](#) class.*
- void [AddLayer](#) (ILayer lay)  
*Adds a layer to the stack.*
- void [RemoveLayer](#) (ILayer lay)  
*Removes a layer from the stack.*
- int [GetSize](#) ()  
*Returns the size of the layerstack.*
- ILayer [GetLayer](#) (int index)  
*Returns the layer at the index.*

### 6.8.1 Beschreibung der Konstruktoren und Destruktoren

#### 6.8.1.1 NSA.Model.NetworkComponents.Layerstack.Layerstack ( )

Initializes a new instance of the [Layerstack](#) class.

### 6.8.2 Dokumentation der Elementfunktionen

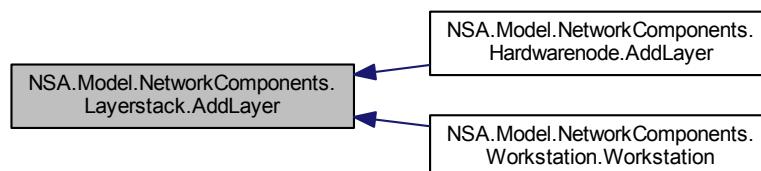
#### 6.8.2.1 void NSA.Model.NetworkComponents.Layerstack.AddLayer ( ILayer lay )

Adds a layer to the stack.

##### Parameter

<i>lay</i>	The layer to be added.
------------	------------------------

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



### 6.8.2.2 ILayer NSA.Model.NetworkComponents.Layerstack.GetLayer ( int *index* )

Returns the layer at the index.

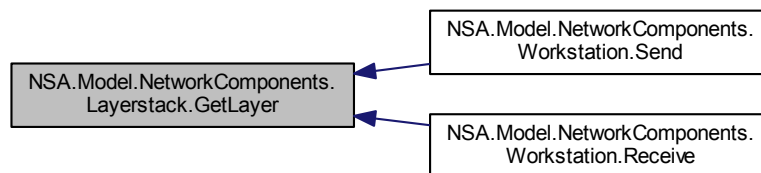
#### Parameter

<i>index</i>	The index.
--------------	------------

#### Rückgabe

The layer

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



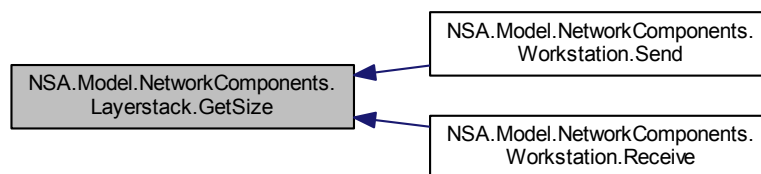
### 6.8.2.3 int NSA.Model.NetworkComponents.Layerstack.GetSize ( )

Returns the size of the layerstack.

#### Rückgabe

The size

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



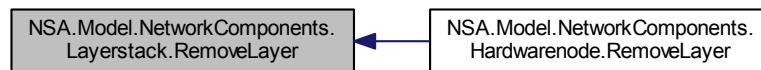
### 6.8.2.4 void NSA.Model.NetworkComponents.Layerstack.RemoveLayer ( ILayer *lay* )

Removes a layer from the stack.

## Parameter

<i>lay</i>	The layer to be removed.
------------	--------------------------

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



Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/[Layerstack.cs](#)

## 6.9 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.*
- void [AddConnection](#) (string StartNodeInterfaceName, string EndNodeInterfaceName, [Connection](#) newConnection)  
*Adds a 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.*

## 6.9.1 Beschreibung der Konstruktoren und Destruktoren

### 6.9.1.1 NSA.Model.NetworkComponents.Network.Network ( )

## 6.9.2 Dokumentation der Elementfunktionen

### 6.9.2.1 void NSA.Model.NetworkComponents.Network.AddConnection ( string *StartNodeInterfaceName*, string *EndNodeInterfaceName*, Connection *newConnection* )

Adds a connection.

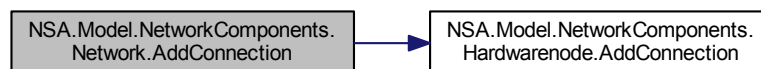
#### Parameter

<i>newConnection</i>	The new connection.
----------------------	---------------------

#### Ausnahmebehandlung

<i>System.InvalidOperationException</i>	<a href="#">Connection</a> already exists!
---	--

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



### 6.9.2.2 void NSA.Model.NetworkComponents.Network.AddHardwarenode ( Hardwarenode *newNode* )

Adds a hardwarenode.

#### Parameter

<i>newNode</i>	The new node.
----------------	---------------

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

Gets all hardwarenodes.

#### Rückgabe

all Hardwarenodes

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

Gets all workstations.

Rückgabe

all Workstations

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

Gets the name of the connection by.

Parameter

<i>Name</i>	The name.
-------------	-----------

Rückgabe

the connection with its name

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

Returns the [Hardwarenode](#) with the name.

Parameter

<i>Name</i>	The name.
-------------	-----------

Rückgabe

The [Hardwarenode](#) with this name or default value

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

Gets the routers with internetconnection.

Rückgabe

A List of routers

**6.9.2.8 Hardwarenode NSA.Model.NetworkComponents.Network.GetWorkstationByIP ( IPAddress ip )**

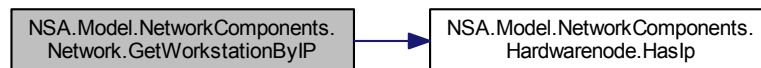
Gets the workstation by ip.

## Parameter

<i>ip</i>	The ip.
-----------	---------

## Rückgabe

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



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

Removes the connection.

## Parameter

<i>ConnectionName</i>	Name of the connection.
-----------------------	-------------------------

#### 6.9.2.10 void NSA.Model.NetworkComponents.Network.RemoveHardwareNode ( string *name* )

Removes the hardwarenode.

## Parameter

<i>name</i>	The name.
-------------	-----------

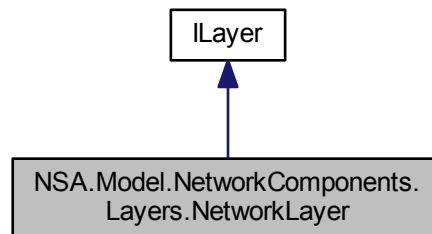
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/[Network.cs](#)

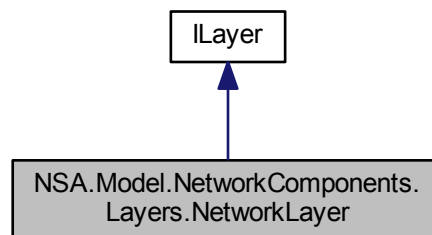


## 6.10 NSA.Model.NetworkComponents.Layers.NetworkLayer Klassenreferenz

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



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



### Öffentliche Methoden

- bool `ValidateReceive` (`Workstation` currentNode, `ValidationInfo` valInfo)  
*Validates the layer while receiving a packet.*
- void `ValidateSend` (`Workstation` destination, `Workstation` currentNode, `ValidationInfo` valInfo)  
*Validates the layer while sending a packet.*

### 6.10.1 Dokumentation der Elementfunktionen

6.10.1.1 bool NSA.Model.NetworkComponents.Layers.NetworkLayer.ValidateReceive ( `Workstation` currentNode, `ValidationInfo` valInfo )

Validates the layer while receiving a packet.

## Parameter

<i>currentNode</i>	
<i>valInfo</i>	

## Rückgabe

Boolean value indicating if the validation was successfull

Implementiert [NSA.Model.NetworkComponents.ILayer](#).

6.10.1.2 void NSA.Model.NetworkComponents.Layers.NetworkLayer.ValidateSend ( Workstation *destination*, Workstation *currentNode*, ValidationInfo *valInfo* )

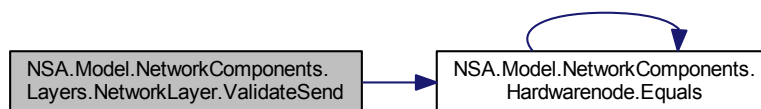
Validates the layer while sending a packet.

## Parameter

<i>destination</i>	The destination.
<i>currentNode</i>	The current node.
<i>valInfo</i>	

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/NetworkComponents/Layers/[NetworkLayer.cs](#)

## 6.11 NSA.Model.BusinessLogic.NodeLocation Klassenreferenz

## Property

- string [Name](#) [get, set]
- Point [Point](#) [get, set]

### 6.11.1 Dokumentation der Propertys

6.11.1.1 `string NSA.Model.BusinessLogic.NodeLocation.Name` `[get]`, `[set]`

6.11.1.2 `Point NSA.Model.BusinessLogic.NodeLocation.Point` `[get]`, `[set]`

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/BusinessLogic/[Project.cs](#)

## 6.12 NSA.Model.BusinessLogic.Packet Klassenreferenz

### Öffentliche Methoden

- [Packet](#) ([Hardwarenode](#) \_source, [Hardwarenode](#) \_destination, int \_ttl, Dictionary< string, object > \_tags, bool expRes)
- [Packet Send](#) ()  
*Sends this packet to the destination.*

### Propertys

- [Result result](#) `[get]`
- bool [expectedResult](#) `[get]`

### 6.12.1 Beschreibung der Konstruktoren und Destruktoren

6.12.1.1 `NSA.Model.BusinessLogic.Packet.Packet ( Hardwarenode _source, Hardwarenode _destination, int _ttl, Dictionary< string, object > _tags, bool expRes )`

### 6.12.2 Dokumentation der Elementfunktionen

6.12.2.1 `Packet NSA.Model.BusinessLogic.Packet.Send ( )`

Sends this packet to the destination.

### Rückgabe

The Returnpacket if sending to destination was successfull

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



### 6.12.3 Dokumentation der Propertys

6.12.3.1 `bool NSA.Model.BusinessLogic.Packet.expectedResult` [get]

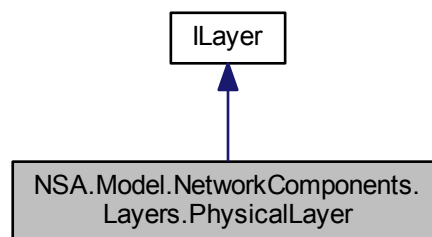
6.12.3.2 `Result NSA.Model.BusinessLogic.Packet.result` [get]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

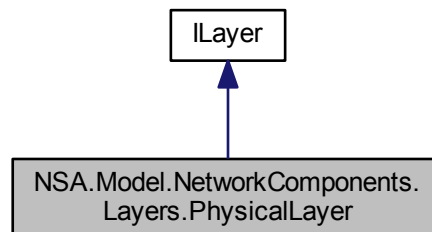
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/BusinessLogic/[Packet.cs](#)

## 6.13 NSA.Model.NetworkComponents.Layers.PhysicalLayer Klassenreferenz

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



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



## Öffentliche Methoden

- bool [ValidateReceive](#) ([Workstation](#) currentNode, [ValidationInfo](#) valInfo)  
*Validates the layer while receiving a packet.*
- void [ValidateSend](#) ([Workstation](#) destination, [Workstation](#) currentNode, [ValidationInfo](#) valInfo)  
*Validates the layer while sending a packet.*

### 6.13.1 Dokumentation der Elementfunktionen

#### 6.13.1.1 bool NSA.Model.NetworkComponents.Layers.PhysicalLayer.ValidateReceive ( [Workstation](#) currentNode, [ValidationInfo](#) valInfo )

Validates the layer while receiving a packet.

##### Parameter

<i>currentNode</i>	
<i>valInfo</i>	

##### Rückgabe

Boolean value indicating if the validation was successfull

Implementiert [NSA.Model.NetworkComponents.ILayer](#).

#### 6.13.1.2 void NSA.Model.NetworkComponents.Layers.PhysicalLayer.ValidateSend ( [Workstation](#) destination, [Workstation](#) currentNode, [ValidationInfo](#) valInfo )

Validates the layer while sending a packet.

##### Parameter

<i>destination</i>	The destination.
<i>currentNode</i>	The current node.
<i>valInfo</i>	

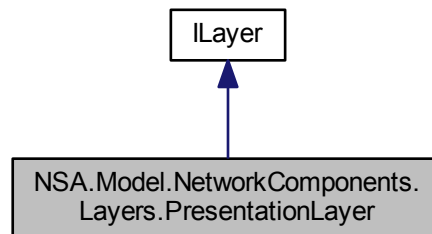
Implementiert [NSA.Model.NetworkComponents.ILayer](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

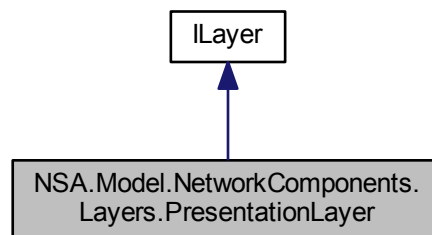
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Layers/[PhysicalLayer.cs](#)

## 6.14 NSA.Model.NetworkComponents.Layers.PresentationLayer Klassenreferenz

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



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



### Öffentliche Methoden

- bool `ValidateReceive` (`Workstation` currentNode, `ValidationInfo` valInfo)  
*Validates the layer while receiving a packet.*
- void `ValidateSend` (`Workstation` destination, `Workstation` currentNode, `ValidationInfo` valInfo)  
*Validates the layer while sending a packet.*

### 6.14.1 Dokumentation der Elementfunktionen

6.14.1.1 bool NSA.Model.NetworkComponents.Layers.PresentationLayer.ValidateReceive ( `Workstation` currentNode, `ValidationInfo` valInfo )

Validates the layer while receiving a packet.

## Parameter

<i>currentNode</i>	
<i>valInfo</i>	

## Rückgabe

Boolean value indicating if the validation was successfull

Implementiert [NSA.Model.NetworkComponents.ILayer](#).

6.14.1.2 void NSA.Model.NetworkComponents.Layers.PresentationLayer.ValidateSend ( Workstation *destination*, Workstation *currentNode*, ValidationInfo *valInfo* )

Validates the layer while sending a packet.

## Parameter

<i>destination</i>	The destination.
<i>currentNode</i>	The current node.
<i>valInfo</i>	

Implementiert [NSA.Model.NetworkComponents.ILayer](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Layers/[PresentationLayer.cs](#)

## 6.15 NSA.Model.BusinessLogic.Project Klassenreferenz

### Öffentliche Methoden

- [Project](#) ()  
*Initializes a new instance of the [Project](#) class.*

### Propertys

- string [Path](#) [get, set]
- [Network Network](#) [get, set]
- List< [NodeLocation](#) > [NodeLocations](#) [get, set]

### 6.15.1 Beschreibung der Konstruktoren und Destruktoren

6.15.1.1 NSA.Model.BusinessLogic.Project.Project ( )

Initializes a new instance of the [Project](#) class.

## 6.15.2 Dokumentation der Propertys

6.15.2.1 **Network** `NSA.Model.BusinessLogic.Project.Network` [get], [set]

6.15.2.2 **List<NodeLocation>** `NSA.Model.BusinessLogic.Project.NodeLocations` [get], [set]

6.15.2.3 **string** `NSA.Model.BusinessLogic.Project.Path` [get], [set]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/BusinessLogic/[Project.cs](#)

## 6.16 NSA.Model.NetworkComponents.Helper\_Classes.Result Klassenreferenz

### Öffentliche Methoden

- [Result](#) ()

### Propertys

- int [ErrorID](#) [get, set]
- string [Res](#) [get, set]
- [ILayer](#) [LayerError](#) [get, set]
- bool [SendError](#) [get, set]

## 6.16.1 Beschreibung der Konstruktoren und Destruktoren

6.16.1.1 **NSA.Model.NetworkComponents.Helper\_Classes.Result.Result** ( )

## 6.16.2 Dokumentation der Propertys

6.16.2.1 **int** `NSA.Model.NetworkComponents.Helper_Classes.Result.ErrorID` [get], [set]

6.16.2.2 **ILayer** `NSA.Model.NetworkComponents.Helper_Classes.Result.LayerError` [get], [set]

6.16.2.3 **string** `NSA.Model.NetworkComponents.Helper_Classes.Result.Res` [get], [set]

6.16.2.4 **bool** `NSA.Model.NetworkComponents.Helper_Classes.Result.SendError` [get], [set]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Helper Classes/[Result.cs](#)



## 6.17 NSA.Model.NetworkComponents.Route Klassenreferenz

### 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]
- IPAddress [Destination](#) [get]
- IPAddress [Subnetmask](#) [get]
- IPAddress [Gateway](#) [get]
- [Interface](#) [Iface](#) [get]

### 6.17.1 Beschreibung der Konstruktoren und Destruktoren

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

Initializes a new instance of the [Route](#) class.

##### Parameter

<i>Destination</i>	The Destination IP.
<i>Subnetmask</i>	The Mask.
<i>Gateway</i>	The Gateway.
<i>Iface</i>	The <a href="#">Interface</a> to be used.

### 6.17.2 Dokumentation der Elementfunktionen

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

Sets the route.

##### Parameter

<i>DestinationIp</i>	The destination.
<i>Mask</i>	The subnetmask.
<i>GatewayAddress</i>	The gateway.
<i>Intface</i>	The iface.

### 6.17.3 Dokumentation der Propertys

6.17.3.1 `IPAddress NSA.Model.NetworkComponents.Route.Destination` `[get]`

6.17.3.2 `IPAddress NSA.Model.NetworkComponents.Route.Gateway` `[get]`

6.17.3.3 `Interface NSA.Model.NetworkComponents.Route.Iface` `[get]`

6.17.3.4 `string NSA.Model.NetworkComponents.Route.Name` `[get]`

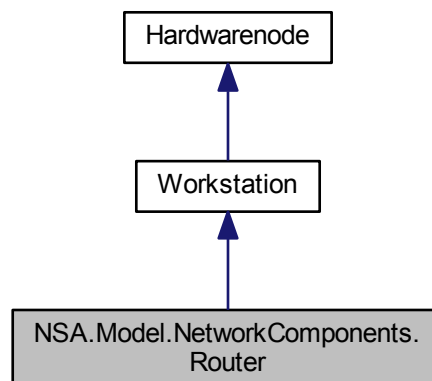
6.17.3.5 `IPAddress NSA.Model.NetworkComponents.Route.Subnetmask` `[get]`

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

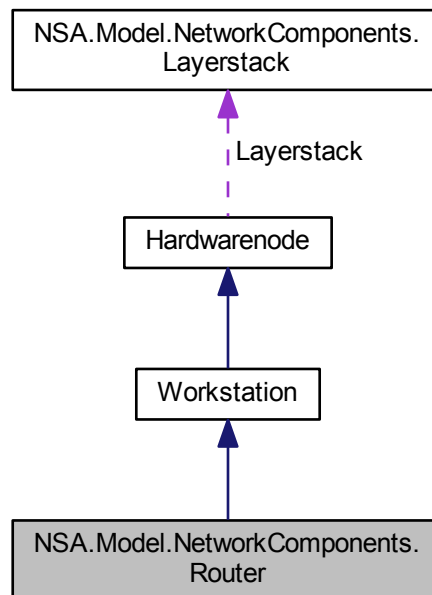
- `C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Route.cs`

## 6.18 NSA.Model.NetworkComponents.Router Klassenreferenz

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]

## Weitere Geerbte Elemente

### 6.18.1 Beschreibung der Konstruktoren und Destruktoren

#### 6.18.1.1 NSA.Model.NetworkComponents.Router.Router ( string *Name* )

Initializes a new instance of the **Router** class.

#### Parameter

<i>Name</i>	The name.
-------------	-----------

## 6.18.2 Dokumentation der Propertys

### 6.18.2.1 bool NSA.Model.NetworkComponents.Router.IsGateway [get], [set]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/[Router.cs](#)

## 6.19 NSA.Model.BusinessLogic.Rule Klassenreferenz

### Öffentliche Methoden

- [Rule](#) ()

### Propertys

- string [Name](#) [get, set]
- List< string > [NodeNames](#) [get, set]
- List< string > [OnlyNodeNames](#) [get, set]
- List< string > [SubnetNames](#) [get, set]
- bool [HasInternet](#) [get, set]
- int [Ttl](#) [get, set]
- bool [Ssl](#) [get, set]
- bool [Applicable](#) [get, set]

### 6.19.1 Beschreibung der Konstruktoren und Destruktoren

#### 6.19.1.1 NSA.Model.BusinessLogic.Rule.Rule ( )

### 6.19.2 Dokumentation der Propertys

#### 6.19.2.1 bool NSA.Model.BusinessLogic.Rule.Applicable [get], [set]

#### 6.19.2.2 bool NSA.Model.BusinessLogic.Rule.HasInternet [get], [set]

#### 6.19.2.3 string NSA.Model.BusinessLogic.Rule.Name [get], [set]

#### 6.19.2.4 List<string> NSA.Model.BusinessLogic.Rule.NodeNames [get], [set]

#### 6.19.2.5 List<string> NSA.Model.BusinessLogic.Rule.OnlyNodeNames [get], [set]

#### 6.19.2.6 bool NSA.Model.BusinessLogic.Rule.Ssl [get], [set]

#### 6.19.2.7 List<string> NSA.Model.BusinessLogic.Rule.SubnetNames [get], [set]

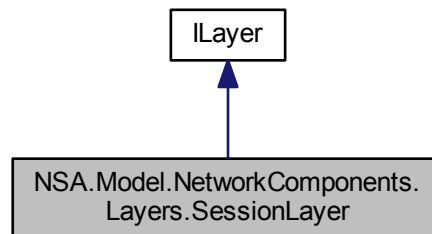
#### 6.19.2.8 int NSA.Model.BusinessLogic.Rule.Ttl [get], [set]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

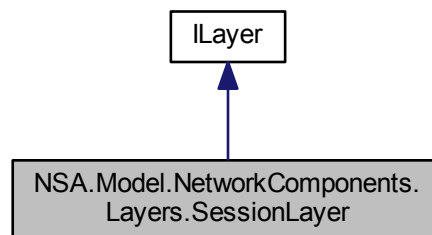
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/BusinessLogic/[Testscenario.cs](#)

## 6.20 NSA.Model.NetworkComponents.Layers.SessionLayer Klassenreferenz

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



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



### Öffentliche Methoden

- `bool ValidateReceive (Workstation currentNode, ValidationInfo valInfo)`  
*Validates the layer while receiving a packet.*
- `void ValidateSend (Workstation destination, Workstation currentNode, ValidationInfo valInfo)`  
*Validates the layer while sending a packet.*

### 6.20.1 Dokumentation der Elementfunktionen

6.20.1.1 `bool NSA.Model.NetworkComponents.Layers.SessionLayer.ValidateReceive ( Workstation currentNode, ValidationInfo valInfo )`

Validates the layer while receiving a packet.

## Parameter

<i>currentNode</i>	
<i>valInfo</i>	

## Rückgabe

Boolean value indicating if the validation was successfull

Implementiert [NSA.Model.NetworkComponents.ILayer](#).

6.20.1.2 void NSA.Model.NetworkComponents.Layers.SessionLayer.ValidateSend ( Workstation *destination*, Workstation *currentNode*, ValidationInfo *valInfo* )

Validates the layer while sending a packet.

## Parameter

<i>destination</i>	The destination.
<i>currentNode</i>	The current node.
<i>valInfo</i>	

Implementiert [NSA.Model.NetworkComponents.ILayer](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Layers/[SessionLayer.cs](#)

## 6.21 NSA.Model.BusinessLogic.Simulation Klassenreferenz

### Öffentliche Methoden

- [Simulation](#) (int *\_id*)
- void [AddPacketSend](#) ([Packet](#) *packet*)  
*Adds the packet send.*
- void [Execute](#) ()  
*Executes this instance.*

### 6.21.1 Beschreibung der Konstruktoren und Destruktoren

6.21.1.1 NSA.Model.BusinessLogic.Simulation.Simulation ( int *\_id* )

### 6.21.2 Dokumentation der Elementfunktionen

6.21.2.1 void NSA.Model.BusinessLogic.Simulation.AddPacketSend ( [Packet](#) *packet* )

Adds the packet send.

## Parameter

<i>packet</i>	The packet.
---------------	-------------

## 6.21.2.2 void NSA.Model.BusinessLogic.Simulation.Execute ( )

Executes this instance.

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

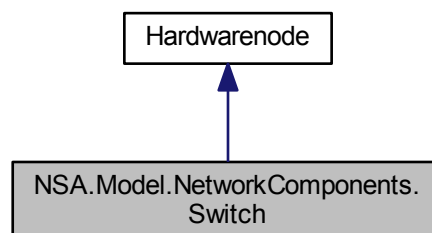


Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

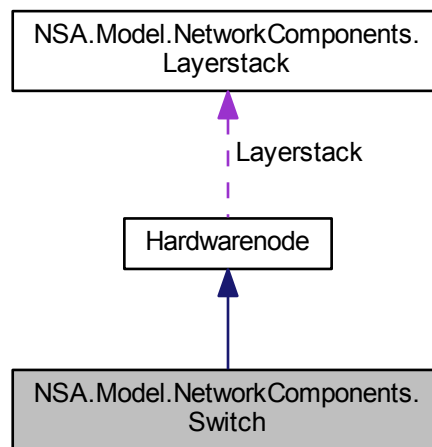
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/BusinessLogic/[Simulation.cs](#)

## 6.22 NSA.Model.NetworkComponents.Switch Klassenreferenz

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.*
- `int GetInterfaceCount ()`  
*Gets the interface count.*
- `string AddInterface ()`  
*Adds the interface.*
- `void RemoveInterface (string InterfaceName)`  
*Removes the interface with the given name.*
- `override List< Hardwarenode > Send (Hardwarenode Destination, Dictionary< string, object > Tags, ValidationInfo valInfo)`  
*Hardwarenode sends the package to specified destination.*
- `bool SendTolp (ValidationInfo valInfo)`

## Propertys

- `List< string > Interfaces [get]`

## Weitere Geerbte Elemente

### 6.22.1 Beschreibung der Konstruktoren und Destruktoren

#### 6.22.1.1 NSA.Model.NetworkComponents.Switch.Switch ( string Name )

Initializes a new instance of the `Switch` class.



## Parameter

<i>Name</i>	The name of the switch.
-------------	-------------------------

## 6.22.2 Dokumentation der Elementfunktionen

### 6.22.2.1 `string NSA.Model.NetworkComponents.Switch.AddInterface ( )`

Adds the interface.

## Rückgabe

string: the name of the new interface

### 6.22.2.2 `int NSA.Model.NetworkComponents.Switch.GetInterfaceCount ( )`

Gets the interface count.

## Rückgabe

int: interface count

### 6.22.2.3 `void NSA.Model.NetworkComponents.Switch.RemoveInterface ( string InterfaceName )`

Removes the interface with the given name.

## Parameter

<i>InterfaceName</i>	The name.
----------------------	-----------

### 6.22.2.4 `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

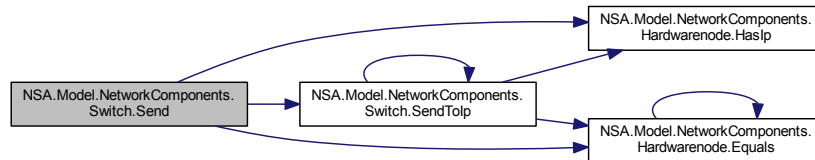
<i>Destination</i>	The destination.
<i>Tags</i>	Optional tags.
<i>valInfo</i>	

## Rückgabe

The [Hardwarenode](#) which received the package or null if an error occurred

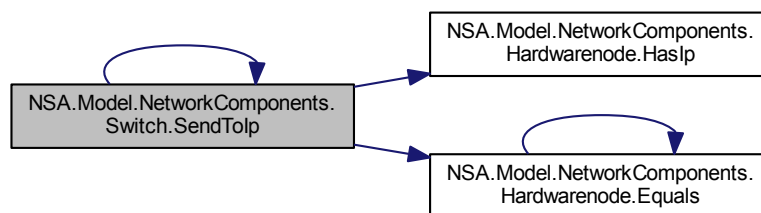
Erneute Implementation von [NSA.Model.NetworkComponents.Hardwarenode](#).

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

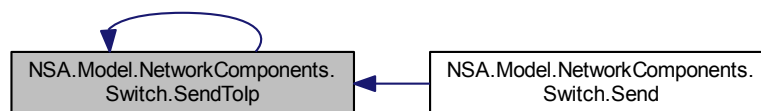


### 6.22.2.5 bool NSA.Model.NetworkComponents.Switch.SendTolp ( ValidationInfo valInfo )

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



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



## 6.22.3 Dokumentation der Propertys

### 6.22.3.1 List<string> NSA.Model.NetworkComponents.Switch.Interfaces [get]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/[Switch.cs](#)

## 6.23 NSA.Model.BusinessLogic.Testscenario Klassenreferenz

### Öffentliche Methoden

- [Testscenario](#) ()
- [Network ParseRulesToNetwork](#) ()

### Propertys

- string [Id](#) [get]
- List< [Rule](#) > [Rules](#) [get, set]

### 6.23.1 Beschreibung der Konstruktoren und Destruktoren

6.23.1.1 [NSA.Model.BusinessLogic.Testscenario.Testscenario](#) ( )

### 6.23.2 Dokumentation der Elementfunktionen

6.23.2.1 [Network NSA.Model.BusinessLogic.Testscenario.ParseRulesToNetwork](#) ( )

### 6.23.3 Dokumentation der Propertys

6.23.3.1 string [NSA.Model.BusinessLogic.Testscenario.Id](#) [get]

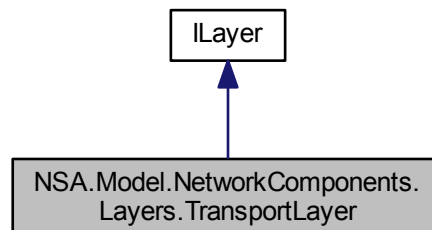
6.23.3.2 List<[Rule](#)> [NSA.Model.BusinessLogic.Testscenario.Rules](#) [get], [set]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

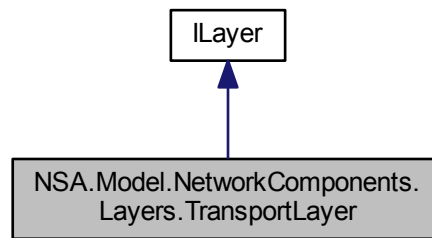
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/BusinessLogic/[Testscenario.cs](#)

## 6.24 NSA.Model.NetworkComponents.Layers.TransportLayer Klassenreferenz

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



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



## Öffentliche Methoden

- bool [ValidateReceive](#) ([Workstation](#) currentNode, [ValidationInfo](#) valInfo)  
*Validates the layer while receiving a packet.*
- void [ValidateSend](#) ([Workstation](#) destination, [Workstation](#) currentNode, [ValidationInfo](#) valInfo)  
*Validates the layer while sending a packet.*

### 6.24.1 Dokumentation der Elementfunktionen

6.24.1.1 bool NSA.Model.NetworkComponents.Layers.TransportLayer.ValidateReceive ( [Workstation](#) *currentNode*, [ValidationInfo](#) *valInfo* )

Validates the layer while receiving a packet.

#### Parameter

<i>currentNode</i>	
<i>valInfo</i>	

#### Rückgabe

Boolean value indicating if the validation was successfull

Implementiert [NSA.Model.NetworkComponents.ILayer](#).

6.24.1.2 void NSA.Model.NetworkComponents.Layers.TransportLayer.ValidateSend ( [Workstation](#) *destination*, [Workstation](#) *currentNode*, [ValidationInfo](#) *valInfo* )

Validates the layer while sending a packet.

## Parameter

<i>destination</i>	The destination.
<i>currentNode</i>	The current node.
<i>valInfo</i>	

Implementiert [NSA.Model.NetworkComponents.ILayer](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Layers/[TransportLayer.cs](#)

## 6.25 NSA.Model.NetworkComponents.Helper\_Classes.ValidationInfo Klassenreferenz

## Property

- List< [Hardwarenode](#) > [NextNodes](#) [get, set]
- IPAddress [NextNodeIP](#) [get, set]
- [Interface](#) [Iface](#) [get, set]
- [Result](#) [Res](#) [get, set]

### 6.25.1 Dokumentation der Property

6.25.1.1 **Interface** [NSA.Model.NetworkComponents.Helper\\_Classes.ValidationInfo.Iface](#) [get], [set]

6.25.1.2 **IPAddress** [NSA.Model.NetworkComponents.Helper\\_Classes.ValidationInfo.NextNodeIP](#) [get], [set]

6.25.1.3 **List<Hardwarenode>** [NSA.Model.NetworkComponents.Helper\\_Classes.ValidationInfo.NextNodes](#) [get], [set]

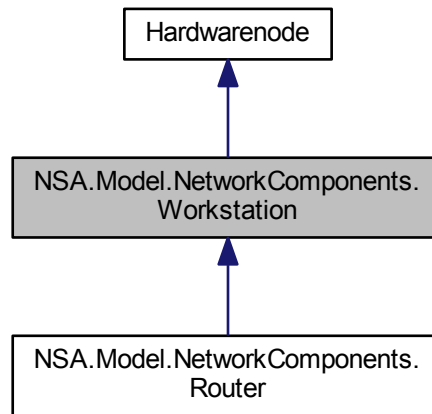
6.25.1.4 **Result** [NSA.Model.NetworkComponents.Helper\\_Classes.ValidationInfo.Res](#) [get], [set]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

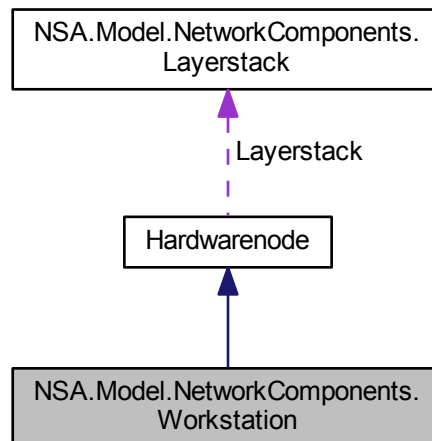
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Helper Classes/[ValidationInfo.cs](#)

## 6.26 NSA.Model.NetworkComponents.Workstation Klassenreferenz

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.*
- `List< Interface > GetInterfaces ()`

- Gets the interfaces.*
- [Interface AddInterface](#) (IPAddress Ip, IPAddress Subnetmask)
  - 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.*
- bool [SetInterface](#) (string Ifacename, IPAddress Ip, IPAddress Mask)
  - Sets the interface.*
- 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](#) receives the package.*

## Propertys

- IPAddress [StandardGateway](#) [get, set]
- [Interface](#) [StandardGatewayPort](#) [get, set]

## Weitere Geerbte Elemente

### 6.26.1 Beschreibung der Konstruktoren und Destruktoren

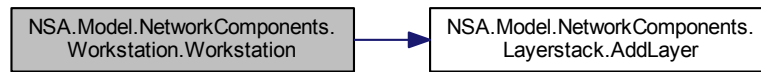
#### 6.26.1.1 NSA.Model.NetworkComponents.Workstation.Workstation ( string Name )

Initializes a new instance of the [Workstation](#) class. The IP address of the standardgateway must be set seperatly.

#### Parameter

<i>Name</i>	The Name.
-------------	-----------

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



## 6.26.2 Dokumentation der Elementfunktionen

### 6.26.2.1 Interface `NSA.Model.NetworkComponents.Workstation.AddInterface ( IPAddress Ip, IPAddress Subnetmask )`

Adds a new interface with the given IP and subnetmask

#### Parameter

<i>Ip</i>	The IP of the interface.
<i>Subnetmask</i>	The subnetmask.

#### Rückgabe

The newly added [Interface](#)

### 6.26.2.2 void `NSA.Model.NetworkComponents.Workstation.AddRoute ( Route Route )`

Adds the route.

#### Parameter

<a href="#">Route</a>	The route.
-----------------------	------------

### 6.26.2.3 int `NSA.Model.NetworkComponents.Workstation.GetInterfaceCount ( )`

Gets the interface count.

#### Rückgabe

int: interface count

### 6.26.2.4 List<Interface> `NSA.Model.NetworkComponents.Workstation.GetInterfaces ( )`

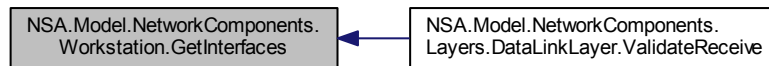
Gets the interfaces.



**Rückgabe**

The Interfaces

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

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

Gets the route at the given index.

**Parameter**

<i>Index</i>	The index.
--------------	------------

**Rückgabe****6.26.2.6 int NSA.Model.NetworkComponents.Workstation.GetRouteCount ( )**

Gets the route count.

**Rückgabe**

int: number of routes in the routingtable

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

Gets the routes.

**Rückgabe**

The Routes

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

Checks if the [Hardwarenode](#) has the IP

## Parameter

<i>Ip</i>	The ip.
-----------	---------

## Rückgabe

bool: true if workstation has the ip, otherwise false

Erneute Implementation von [NSA.Model.NetworkComponents.Hardwarenode](#).

6.26.2.9 override bool NSA.Model.NetworkComponents.Workstation.Receive ( Dictionary< string, object > *Tags*, ValidationInfo *valInfo* ) [virtual]

[Hardwarenode](#) receives the package.

## Parameter

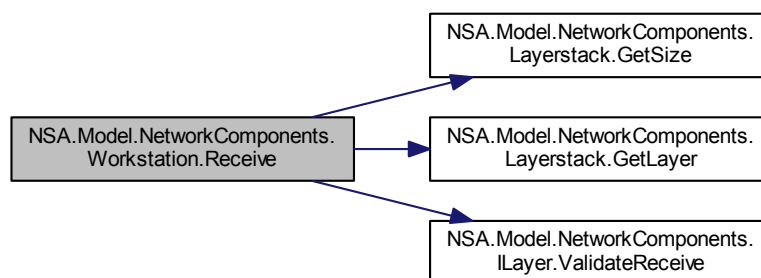
<i>Tags</i>	Optional tags.
<i>Res</i>	
<i>Next↔ NodeIp</i>	

## 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:



6.26.2.10 void NSA.Model.NetworkComponents.Workstation.RemoveInterface ( string *InterfaceName* )

Removes the interface with the given name.

## Parameter

<i>InterfaceName</i>	The Interfacename.
----------------------	--------------------

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

Removes the route.

## Parameter

<i>N</i>	The name.
----------	-----------

6.26.2.12 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

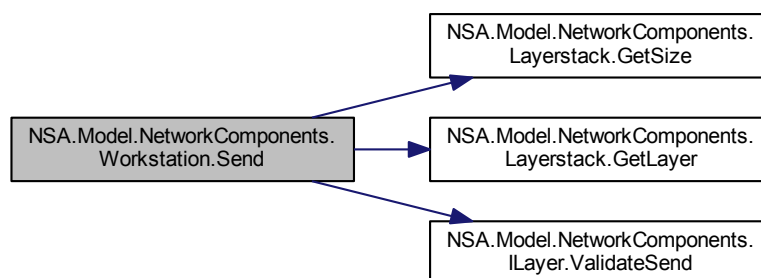
<i>Destination</i>	The destination.
<i>Tags</i>	Optional tags.
<i>valInfo</i>	

## Rückgabe

The [Hardwarenode](#) which received the package or null if an error occurred

Erneute Implementation von [NSA.Model.NetworkComponents.Hardwarenode](#).

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

6.26.2.13 bool NSA.Model.NetworkComponents.Workstation.SetInterface ( string *Ifacename*, IPAddress *Ip*, IPAddress *Mask* )

Sets the interface.

## Parameter

<i>Ifacename</i>	The name of the <a href="#">Interface</a> .
<i>Ip</i>	The new ip.
<i>Mask</i>	The new subnetmask.

## Rückgabe

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

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

Sets the route.

## Parameter

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

## Rückgabe

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

## 6.26.3 Dokumentation der Propertys

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

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

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](#)

#### 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 [NSA.Model.BusinessLogic.NodeLocation](#)

#### Namensbereiche

- namespace [NSA.Model.BusinessLogic](#)

### 7.3 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/BusinessLogic/Simulation.cs-Dateireferenz

#### Klassen

- class [NSA.Model.BusinessLogic.Simulation](#)

## Namensbereiche

- namespace [NSA.Model.BusinessLogic](#)

## 7.4 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/BusinessLogic/Testscenario.cs-Dateireferenz

### Klassen

- class [NSA.Model.BusinessLogic.Testscenario](#)
- class [NSA.Model.BusinessLogic.Rule](#)

## Namensbereiche

- namespace [NSA.Model.BusinessLogic](#)

## 7.5 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Connection.cs-Dateireferenz

### Klassen

- class [NSA.Model.NetworkComponents.Connection](#)

## Namensbereiche

- namespace [NSA.Model.NetworkComponents](#)

## 7.6 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Hardwarenode.cs-Dateireferenz

### Klassen

- class [NSA.Model.NetworkComponents.Hardwarenode](#)

## Namensbereiche

- namespace [NSA.Model.NetworkComponents](#)

## 7.7 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Helper Classes/IPAddressExtensions.cs-Dateireferenz

### Klassen

- class [NSA.Model.NetworkComponents.Helper\\_Classes.IPAddressExtensions](#)

- namespace [NSA.Model.NetworkComponents.Helper\\_Classes](#)

## 7.8 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Helper Classes/Result.cs-Dateireferenz

### Klassen

- class [NSA.Model.NetworkComponents.Helper\\_Classes.Result](#)

### Namensbereiche

- namespace [NSA.Model.NetworkComponents.Helper\\_Classes](#)

## 7.9 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Helper Classes/ValidationInfo.cs-Dateireferenz

### Klassen

- class [NSA.Model.NetworkComponents.Helper\\_Classes.ValidationInfo](#)

### Namensbereiche

- namespace [NSA.Model.NetworkComponents.Helper\\_Classes](#)

## 7.10 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/ILayer.cs-Dateireferenz

### Klassen

- interface [NSA.Model.NetworkComponents.ILayer](#)

### Namensbereiche

- namespace [NSA.Model.NetworkComponents](#)

## 7.11 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Interface.cs-Dateireferenz

### Klassen

- class [NSA.Model.NetworkComponents.Interface](#)

## Namensbereiche

- namespace [NSA.Model.NetworkComponents](#)

### 7.12 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Layers/ApplicationLayer.cs-Dateireferenz

## Klassen

- class [NSA.Model.NetworkComponents.Layers.ApplicationLayer](#)

## Namensbereiche

- namespace [NSA.Model.NetworkComponents.Layers](#)

### 7.13 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Layers/DataLinkLayer.cs-Dateireferenz

## Klassen

- class [NSA.Model.NetworkComponents.Layers.DataLinkLayer](#)

## Namensbereiche

- namespace [NSA.Model.NetworkComponents.Layers](#)

### 7.14 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Layers/NetworkLayer.cs-Dateireferenz

## Klassen

- class [NSA.Model.NetworkComponents.Layers.NetworkLayer](#)

## Namensbereiche

- namespace [NSA.Model.NetworkComponents.Layers](#)

### 7.15 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Layers/PhysicalLayer.cs-Dateireferenz

## Klassen

- class [NSA.Model.NetworkComponents.Layers.PhysicalLayer](#)



#### Namensbereiche

- namespace [NSA.Model.NetworkComponents.Layers](#)

### 7.16 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Layers/PresentationLayer.cs-Dateireferenz

#### Klassen

- class [NSA.Model.NetworkComponents.Layers.PresentationLayer](#)

#### Namensbereiche

- namespace [NSA.Model.NetworkComponents.Layers](#)

### 7.17 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Layers/SessionLayer.cs-Dateireferenz

#### Klassen

- class [NSA.Model.NetworkComponents.Layers.SessionLayer](#)

#### Namensbereiche

- namespace [NSA.Model.NetworkComponents.Layers](#)

### 7.18 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Layers/TransportLayer.cs-Dateireferenz

#### Klassen

- class [NSA.Model.NetworkComponents.Layers.TransportLayer](#)

#### Namensbereiche

- namespace [NSA.Model.NetworkComponents.Layers](#)

### 7.19 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Layerstack.cs-Dateireferenz

#### Klassen

- class [NSA.Model.NetworkComponents.Layerstack](#)

## Namensbereiche

- namespace [NSA.Model.NetworkComponents](#)

## 7.20 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Network.cs-Dateireferenz

### Klassen

- class [NSA.Model.NetworkComponents.Network](#)

## Namensbereiche

- namespace [NSA.Model.NetworkComponents](#)

## 7.21 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Route.cs-Dateireferenz

### Klassen

- class [NSA.Model.NetworkComponents.Route](#)

## Namensbereiche

- namespace [NSA.Model.NetworkComponents](#)

## 7.22 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Router.cs-Dateireferenz

### Klassen

- class [NSA.Model.NetworkComponents.Router](#)

## Namensbereiche

- namespace [NSA.Model.NetworkComponents](#)

## 7.23 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↔ NSA.Model/NetworkComponents/Switch.cs-Dateireferenz

### Klassen

- class [NSA.Model.NetworkComponents.Switch](#)

#### Namensbereiche

- namespace [NSA.Model.NetworkComponents](#)

#### 7.24 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↵ NSA.Model/NetworkComponents/Workstation.cs-Dateireferenz

#### Klassen

- class [NSA.Model.NetworkComponents.Workstation](#)

#### Namensbereiche

- namespace [NSA.Model.NetworkComponents](#)

#### 7.25 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↵ NSA.Model/obj/Debug/TemporaryGeneratedFile\_036C0B5B-1481-4323-8D20-8F5A↵ DCB23D92.cs-Dateireferenz

#### 7.26 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↵ NSA.Model/obj/Debug/TemporaryGeneratedFile\_5937a670-0e60-4077-877b-f7221da3dda1.cs- Dateireferenz

#### 7.27 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↵ NSA.Model/obj/Debug/TemporaryGeneratedFile\_E7A71F73-0F8D-4B9B-B56E-8↵ E70B10BC5D3.cs-Dateireferenz

#### 7.28 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↵ NSA.Model/Properties/AssemblyInfo.cs-Dateireferenz



# Index

AddConnection  
    NSA::Model::NetworkComponents::Hardwarenode, 20  
    NSA::Model::NetworkComponents::Network, 34  
AddHardwarenode  
    NSA::Model::NetworkComponents::Network, 34  
AddInterface  
    NSA::Model::NetworkComponents::Switch, 53  
    NSA::Model::NetworkComponents::Workstation, 60  
AddLayer  
    NSA::Model::NetworkComponents::Hardwarenode, 20  
    NSA::Model::NetworkComponents::Layerstack, 31  
AddPacketSend  
    NSA::Model::BusinessLogic::Simulation, 50  
AddRoute  
    NSA::Model::NetworkComponents::Workstation, 60  
Applicable  
    NSA::Model::BusinessLogic::Rule, 48  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    BusinessLogic/Packet.cs, 65  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    BusinessLogic/Project.cs, 65  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    BusinessLogic/Simulation.cs, 65  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    BusinessLogic/Testscenario.cs, 66  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    NetworkComponents/Connection.cs, 66  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    NetworkComponents/Hardwarenode.cs, 66  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    NetworkComponents/Helper    Classes/IP↔  
    AddressExtensions.cs, 66  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    NetworkComponents/Helper    Classes/↔  
    Result.cs, 67  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
NetworkComponents/Helper    Classes/↔  
    ValidationInfo.cs, 67  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    NetworkComponents/ILayer.cs, 67  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    NetworkComponents/Interface.cs, 67  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    NetworkComponents/Layers/Application↔  
    Layer.cs, 68  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    NetworkComponents/Layers/DataLink↔  
    Layer.cs, 68  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    NetworkComponents/Layers/NetworkLayer.↔  
    cs, 68  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    NetworkComponents/Layers/PhysicalLayer.↔  
    cs, 68  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    NetworkComponents/Layers/Presentation↔  
    Layer.cs, 69  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    NetworkComponents/Layers/SessionLayer.↔  
    cs, 69  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    NetworkComponents/Layers/Transport↔  
    Layer.cs, 69  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    NetworkComponents/Layerstack.cs, 69  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    NetworkComponents/Network.cs, 70  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    NetworkComponents/Route.cs, 70  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
    NetworkComponents/Router.cs, 70  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔

- NetworkSimulatorAnalyzer/NSA.Model/↔
- NetworkComponents/Switch.cs, 70
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
- NetworkSimulatorAnalyzer/NSA.Model/↔
- NetworkComponents/Workstation.cs, 71
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
- NetworkSimulatorAnalyzer/NSA.Model/↔
- Properties/AssemblyInfo.cs, 71
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
- NetworkSimulatorAnalyzer/NSA.Model/obj/↔
- Debug/TemporaryGeneratedFile\_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs, 71
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
- NetworkSimulatorAnalyzer/NSA.Model/obj/↔
- Debug/TemporaryGeneratedFile\_5937a670-0e60-4077-877b-f7221da3dda1.cs, 71
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
- NetworkSimulatorAnalyzer/NSA.Model/obj/↔
- Debug/TemporaryGeneratedFile\_E7A71F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs, 71
- Connection
  - NSA::Model::NetworkComponents::Connection, 13
- Connections
  - NSA::Model::NetworkComponents::Hardwarenode, 25
- Destination
  - NSA::Model::NetworkComponents::Route, 46
- End
  - NSA::Model::NetworkComponents::Connection, 15
- Equals
  - NSA::Model::NetworkComponents::Connection, 13, 14
  - NSA::Model::NetworkComponents::Hardwarenode, 21, 22
- ErrorID
  - NSA::Model::NetworkComponents::Helper\_↔
  - Classes::Result, 44
- Execute
  - NSA::Model::BusinessLogic::Simulation, 51
- expectedResult
  - NSA::Model::BusinessLogic::Packet, 40
- Gateway
  - NSA::Model::NetworkComponents::Route, 46
- GetAllHardwarenodes
  - NSA::Model::NetworkComponents::Network, 34
- GetAllWorkstations
  - NSA::Model::NetworkComponents::Network, 34
- GetBroadcastAddress
  - NSA::Model::NetworkComponents::Helper\_↔
  - Classes::IPAddressExtensions, 30
- GetConnectionByName
  - NSA::Model::NetworkComponents::Network, 35
- GetConnections
  - NSA::Model::NetworkComponents::Hardwarenode, 22
- GetHardwarenodeByName
  - NSA::Model::NetworkComponents::Network, 35
- GetHashCode
  - NSA::Model::NetworkComponents::Connection, 14
  - NSA::Model::NetworkComponents::Hardwarenode, 22
- GetInterfaceCount
  - NSA::Model::NetworkComponents::Switch, 53
  - NSA::Model::NetworkComponents::Workstation, 60
- GetInterfaces
  - NSA::Model::NetworkComponents::Workstation, 60
- GetLayer
  - NSA::Model::NetworkComponents::Layerstack, 31
- GetNetworkAddress
  - NSA::Model::NetworkComponents::Helper\_↔
  - Classes::IPAddressExtensions, 30
- GetPortIndex
  - NSA::Model::NetworkComponents::Connection, 14
- GetPortIndexOfConnection
  - NSA::Model::NetworkComponents::Hardwarenode, 22
- GetRouteAt
  - NSA::Model::NetworkComponents::Workstation, 61
- GetRouteCount
  - NSA::Model::NetworkComponents::Workstation, 61
- GetRouters
  - NSA::Model::NetworkComponents::Network, 35
- GetRoutes
  - NSA::Model::NetworkComponents::Workstation, 61
- GetSize
  - NSA::Model::NetworkComponents::Layerstack, 32
- GetWorkstationByIP
  - NSA::Model::NetworkComponents::Network, 35
- Hardwarenode
  - NSA::Model::NetworkComponents::Hardwarenode, 20
- HasInternet
  - NSA::Model::BusinessLogic::Rule, 48
- HasIp
  - NSA::Model::NetworkComponents::Hardwarenode, 23
  - NSA::Model::NetworkComponents::Workstation, 61
- Id
  - NSA::Model::BusinessLogic::Testscenario, 55
- Iface
  - NSA::Model::NetworkComponents::Helper\_↔
  - Classes::ValidationInfo, 57
  - NSA::Model::NetworkComponents::Route, 46
- Interface
  - NSA::Model::NetworkComponents::Interface, 29
- Interfaces
  - NSA::Model::NetworkComponents::Switch, 54

- IpAddress
  - NSA::Model::NetworkComponents::Interface, [30](#)
- IsGateway
  - NSA::Model::NetworkComponents::Router, [48](#)
- IsInSameSubnet
  - NSA::Model::NetworkComponents::Helper\_↔
    - Classes::IPAddressExtensions, [30](#)
- LayerError
  - NSA::Model::NetworkComponents::Helper\_↔
    - Classes::Result, [44](#)
- Layerstack
  - NSA::Model::NetworkComponents::Hardwarenode, [25](#)
  - NSA::Model::NetworkComponents::Layerstack, [31](#)
- NSA.Model, [9](#)
- NSA.Model.BusinessLogic, [9](#)
- NSA.Model.BusinessLogic.NodeLocation, [38](#)
- NSA.Model.BusinessLogic.Packet, [39](#)
- NSA.Model.BusinessLogic.Project, [43](#)
- NSA.Model.BusinessLogic.Rule, [48](#)
- NSA.Model.BusinessLogic.Simulation, [50](#)
- NSA.Model.BusinessLogic.Testscenario, [55](#)
- NSA.Model.NetworkComponents, [9](#)
- NSA.Model.NetworkComponents.Connection, [12](#)
- NSA.Model.NetworkComponents.Hardwarenode, [18](#)
- NSA.Model.NetworkComponents.Helper\_Classes, [10](#)
- NSA.Model.NetworkComponents.Helper\_Classes.IP↔
  - AddressExtensions, [30](#)
- NSA.Model.NetworkComponents.Helper\_Classes.↔
  - Result, [44](#)
- NSA.Model.NetworkComponents.Helper\_Classes.↔
  - ValidationInfo, [57](#)
- NSA.Model.NetworkComponents.ILayer, [26](#)
- NSA.Model.NetworkComponents.Interface, [29](#)
- NSA.Model.NetworkComponents.Layers, [10](#)
- NSA.Model.NetworkComponents.Layers.Application↔
  - Layer, [11](#)
- NSA.Model.NetworkComponents.Layers.DataLink↔
  - Layer, [16](#)
- NSA.Model.NetworkComponents.Layers.NetworkLayer, [37](#)
- NSA.Model.NetworkComponents.Layers.PhysicalLayer, [40](#)
- NSA.Model.NetworkComponents.Layers.Presentation↔
  - Layer, [42](#)
- NSA.Model.NetworkComponents.Layers.SessionLayer, [49](#)
- NSA.Model.NetworkComponents.Layers.Transport↔
  - Layer, [55](#)
- NSA.Model.NetworkComponents.Layerstack, [31](#)
- NSA.Model.NetworkComponents.Network, [33](#)
- NSA.Model.NetworkComponents.Route, [45](#)
- NSA.Model.NetworkComponents.Router, [46](#)
- NSA.Model.NetworkComponents.Switch, [51](#)
- NSA.Model.NetworkComponents.Workstation, [58](#)
- NSA::Model::BusinessLogic::NodeLocation
  - Name, [39](#)
- Point, [39](#)
- NSA::Model::BusinessLogic::Packet
  - expectedResult, [40](#)
  - Packet, [39](#)
  - result, [40](#)
  - Send, [39](#)
- NSA::Model::BusinessLogic::Project
  - Network, [44](#)
  - NodeLocations, [44](#)
  - Path, [44](#)
  - Project, [43](#)
- NSA::Model::BusinessLogic::Rule
  - Applicable, [48](#)
  - HasInternet, [48](#)
  - Name, [48](#)
  - NodeNames, [48](#)
  - OnlyNodeNames, [48](#)
  - Rule, [48](#)
  - Ssl, [48](#)
  - SubnetNames, [48](#)
  - Ttl, [48](#)
- NSA::Model::BusinessLogic::Simulation
  - AddPacketSend, [50](#)
  - Execute, [51](#)
  - Simulation, [50](#)
- NSA::Model::BusinessLogic::Testscenario
  - Id, [55](#)
  - ParseRulesToNetwork, [55](#)
  - Rules, [55](#)
  - Testscenario, [55](#)
- NSA::Model::NetworkComponents::Connection
  - Connection, [13](#)
  - End, [15](#)
  - Equals, [13](#), [14](#)
  - GetHashCode, [14](#)
  - GetPortIndex, [14](#)
  - Name, [15](#)
  - operator!=, [15](#)
  - operator==, [15](#)
  - Start, [16](#)
- NSA::Model::NetworkComponents::Hardwarenode
  - AddConnection, [20](#)
  - AddLayer, [20](#)
  - Connections, [25](#)
  - Equals, [21](#), [22](#)
  - GetConnections, [22](#)
  - GetHashCode, [22](#)
  - GetPortIndexOfConnection, [22](#)
  - Hardwarenode, [20](#)
  - HasIp, [23](#)
  - Layerstack, [25](#)
  - Name, [26](#)
  - operator!=, [23](#)
  - operator==, [24](#)
  - Receive, [24](#)
  - RemoveConnection, [24](#)
  - RemoveLayer, [25](#)
  - Send, [25](#)

- NSA::Model::NetworkComponents::Helper\_Classes::I↔
  - PAddressExtensions
  - GetBroadcastAddress, 30
  - GetNetworkAddress, 30
  - IsInSameSubnet, 30
- NSA::Model::NetworkComponents::Helper\_Classes::↔
  - Result
  - ErrorID, 44
  - LayerError, 44
  - Res, 44
  - Result, 44
  - SendError, 44
- NSA::Model::NetworkComponents::Helper\_Classes::↔
  - ValidationInfo
  - Iface, 57
  - NextNodeIP, 57
  - NextNodes, 57
  - Res, 57
- NSA::Model::NetworkComponents::ILayer
  - ValidateReceive, 27
  - ValidateSend, 28
- NSA::Model::NetworkComponents::Interface
  - Interface, 29
  - IpAddress, 30
  - Name, 30
  - NamePrefix, 30
  - SetInterface, 29
  - Subnetmask, 30
- NSA::Model::NetworkComponents::Layers::Application↔
  - Layer
  - ValidateReceive, 12
  - ValidateSend, 12
- NSA::Model::NetworkComponents::Layers::DataLink↔
  - Layer
  - ValidateReceive, 17
  - ValidateSend, 17
- NSA::Model::NetworkComponents::Layers::Network↔
  - Layer
  - ValidateReceive, 37
  - ValidateSend, 38
- NSA::Model::NetworkComponents::Layers::Physical↔
  - Layer
  - ValidateReceive, 41
  - ValidateSend, 41
- NSA::Model::NetworkComponents::Layers::Presentation↔
  - Layer
  - ValidateReceive, 42
  - ValidateSend, 43
- NSA::Model::NetworkComponents::Layers::Session↔
  - Layer
  - ValidateReceive, 49
  - ValidateSend, 50
- NSA::Model::NetworkComponents::Layers::Transport↔
  - Layer
  - ValidateReceive, 56
  - ValidateSend, 56
- NSA::Model::NetworkComponents::Layerstack
  - AddLayer, 31
  - GetLayer, 31
  - GetSize, 32
  - Layerstack, 31
  - RemoveLayer, 32
- NSA::Model::NetworkComponents::Network
  - AddConnection, 34
  - AddHardwarenode, 34
  - GetAllHardwarenodes, 34
  - GetAllWorkstations, 34
  - GetConnectionByName, 35
  - GetHardwarenodeByName, 35
  - GetRouters, 35
  - GetWorkstationByIP, 35
  - Network, 34
  - RemoveConnection, 36
  - RemoveHardwarnode, 36
- NSA::Model::NetworkComponents::Route
  - Destination, 46
  - Gateway, 46
  - Iface, 46
  - Name, 46
  - Route, 45
  - SetRoute, 45
  - Subnetmask, 46
- NSA::Model::NetworkComponents::Router
  - IsGateway, 48
  - Router, 47
- NSA::Model::NetworkComponents::Switch
  - AddInterface, 53
  - GetInterfaceCount, 53
  - Interfaces, 54
  - RemoveInterface, 53
  - Send, 53
  - SendToIp, 54
  - Switch, 52
- NSA::Model::NetworkComponents::Workstation
  - AddInterface, 60
  - AddRoute, 60
  - GetInterfaceCount, 60
  - GetInterfaces, 60
  - GetRouteAt, 61
  - GetRouteCount, 61
  - GetRoutes, 61
  - HasIp, 61
  - Receive, 62
  - RemoveInterface, 62
  - RemoveRoute, 63
  - Send, 63
  - SetInterface, 63
  - SetRoute, 64
  - StandardGateway, 64
  - StandardGatewayPort, 64
  - Workstation, 59
- NSA, 9
- Name
  - NSA::Model::BusinessLogic::NodeLocation, 39
  - NSA::Model::BusinessLogic::Rule, 48
  - NSA::Model::NetworkComponents::Connection, 15



- NSA::Model::NetworkComponents::Hardwarenode, 26
- NSA::Model::NetworkComponents::Interface, 30
- NSA::Model::NetworkComponents::Route, 46
- NamePrefix
  - NSA::Model::NetworkComponents::Interface, 30
- Network
  - NSA::Model::BusinessLogic::Project, 44
  - NSA::Model::NetworkComponents::Network, 34
- NextNodeIP
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::ValidationInfo, 57
- NextNodes
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::ValidationInfo, 57
- NodeLocations
  - NSA::Model::BusinessLogic::Project, 44
- NodeNames
  - NSA::Model::BusinessLogic::Rule, 48
- OnlyNodeNames
  - NSA::Model::BusinessLogic::Rule, 48
- operator!=
  - NSA::Model::NetworkComponents::Connection, 15
  - NSA::Model::NetworkComponents::Hardwarenode, 23
- operator==
  - NSA::Model::NetworkComponents::Connection, 15
  - NSA::Model::NetworkComponents::Hardwarenode, 24
- Packet
  - NSA::Model::BusinessLogic::Packet, 39
- ParseRulesToNetwork
  - NSA::Model::BusinessLogic::Testscenario, 55
- Path
  - NSA::Model::BusinessLogic::Project, 44
- Point
  - NSA::Model::BusinessLogic::NodeLocation, 39
- Project
  - NSA::Model::BusinessLogic::Project, 43
- Receive
  - NSA::Model::NetworkComponents::Hardwarenode, 24
  - NSA::Model::NetworkComponents::Workstation, 62
- RemoveConnection
  - NSA::Model::NetworkComponents::Hardwarenode, 24
  - NSA::Model::NetworkComponents::Network, 36
- RemoveHardwarnode
  - NSA::Model::NetworkComponents::Network, 36
- RemoveInterface
  - NSA::Model::NetworkComponents::Switch, 53
  - NSA::Model::NetworkComponents::Workstation, 62
- RemoveLayer
  - NSA::Model::NetworkComponents::Hardwarenode, 25
  - NSA::Model::NetworkComponents::Layerstack, 32
- RemoveRoute
  - NSA::Model::NetworkComponents::Workstation, 63
- Res
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::Result, 44
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::ValidationInfo, 57
- Result
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::Result, 44
- result
  - NSA::Model::BusinessLogic::Packet, 40
- Route
  - NSA::Model::NetworkComponents::Route, 45
- Router
  - NSA::Model::NetworkComponents::Router, 47
- Rule
  - NSA::Model::BusinessLogic::Rule, 48
- Rules
  - NSA::Model::BusinessLogic::Testscenario, 55
- Send
  - NSA::Model::BusinessLogic::Packet, 39
  - NSA::Model::NetworkComponents::Hardwarenode, 25
  - NSA::Model::NetworkComponents::Switch, 53
  - NSA::Model::NetworkComponents::Workstation, 63
- SendError
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::Result, 44
- SendToIp
  - NSA::Model::NetworkComponents::Switch, 54
- SetInterface
  - NSA::Model::NetworkComponents::Interface, 29
  - NSA::Model::NetworkComponents::Workstation, 63
- SetRoute
  - NSA::Model::NetworkComponents::Route, 45
  - NSA::Model::NetworkComponents::Workstation, 64
- Simulation
  - NSA::Model::BusinessLogic::Simulation, 50
- Ssl
  - NSA::Model::BusinessLogic::Rule, 48
- StandardGateway
  - NSA::Model::NetworkComponents::Workstation, 64
- StandardGatewayPort
  - NSA::Model::NetworkComponents::Workstation, 64
- Start
  - NSA::Model::NetworkComponents::Connection, 16
- SubnetNames
  - NSA::Model::BusinessLogic::Rule, 48

## Subnetmask

NSA::Model::NetworkComponents::Interface, 30

NSA::Model::NetworkComponents::Route, 46

## Switch

NSA::Model::NetworkComponents::Switch, 52

## Testscenario

NSA::Model::BusinessLogic::Testscenario, 55

## Ttl

NSA::Model::BusinessLogic::Rule, 48

## ValidateReceive

NSA::Model::NetworkComponents::ILayer, 27

NSA::Model::NetworkComponents::Layers::↔  
ApplicationLayer, 12NSA::Model::NetworkComponents::Layers::Data↔  
LinkLayer, 17NSA::Model::NetworkComponents::Layers::↔  
NetworkLayer, 37NSA::Model::NetworkComponents::Layers::↔  
PhysicalLayer, 41NSA::Model::NetworkComponents::Layers::↔  
PresentationLayer, 42NSA::Model::NetworkComponents::Layers::↔  
SessionLayer, 49NSA::Model::NetworkComponents::Layers::↔  
TransportLayer, 56

## ValidateSend

NSA::Model::NetworkComponents::ILayer, 28

NSA::Model::NetworkComponents::Layers::↔  
ApplicationLayer, 12NSA::Model::NetworkComponents::Layers::Data↔  
LinkLayer, 17NSA::Model::NetworkComponents::Layers::↔  
NetworkLayer, 38NSA::Model::NetworkComponents::Layers::↔  
PhysicalLayer, 41NSA::Model::NetworkComponents::Layers::↔  
PresentationLayer, 43NSA::Model::NetworkComponents::Layers::↔  
SessionLayer, 50NSA::Model::NetworkComponents::Layers::↔  
TransportLayer, 56

## Workstation

NSA::Model::NetworkComponents::Workstation,  
59