

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
5.3.1	Dokumentation der Aufzählungstypen . . . . .	10
5.3.1.1	SimulationType . . . . .	10
5.4	NSA.Model.BusinessLogic.TestscenarioRunnables-Namensbereichsreferenz . . . . .	10
5.5	NSA.Model.NetworkComponents-Namensbereichsreferenz . . . . .	10
5.6	NSA.Model.NetworkComponents.Helper_Classes-Namensbereichsreferenz . . . . .	11
5.7	NSA.Model.NetworkComponents.Layers-Namensbereichsreferenz . . . . .	11

<b>6</b>	<b>Klassen-Dokumentation</b>	<b>13</b>
6.1	NSA.Model.NetworkComponents.Layers.ApplicationLayer Klassenreferenz	13
6.1.1	Ausführliche Beschreibung	14
6.1.2	Beschreibung der Konstruktoren und Destruktoren	14
6.1.2.1	ApplicationLayer(int l)	14
6.1.3	Dokumentation der Elementfunktionen	14
6.1.3.1	GetLayerIndex()	14
6.1.3.2	GetLayerName()	15
6.1.3.3	SetLayerIndex(int l)	15
6.1.3.4	SetLayerName(string NewName)	15
6.1.3.5	ValidateReceive(Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)	15
6.1.3.6	ValidateSend(Workstation Destination, Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, int LayerIndex)	16
6.2	NSA.Model.NetworkComponents.Connection Klassenreferenz	16
6.2.1	Ausführliche Beschreibung	17
6.2.2	Beschreibung der Konstruktoren und Destruktoren	17
6.2.2.1	Connection(Hardwarenode Source, Hardwarenode Target)	17
6.2.3	Dokumentation der Elementfunktionen	17
6.2.3.1	Equals(object Obj)	17
6.2.3.2	Equals(Connection Other)	18
6.2.3.3	GetHashCode()	18
6.2.3.4	GetPortIndex(Hardwarenode Node)	18
6.2.3.5	operator!=(Connection A, Connection B)	18
6.2.3.6	operator==(Connection A, Connection B)	19
6.2.4	Dokumentation der Propertys	19
6.2.4.1	End	19
6.2.4.2	Name	19
6.2.4.3	Start	20
6.3	NSA.Model.NetworkComponents.Layers.CustomLayer Klassenreferenz	20
6.3.1	Ausführliche Beschreibung	21

6.3.2	Beschreibung der Konstruktoren und Destruktoren	21
6.3.2.1	CustomLayer(string N, int I)	21
6.3.3	Dokumentation der Elementfunktionen	21
6.3.3.1	GetLayerIndex()	21
6.3.3.2	GetLayerName()	22
6.3.3.3	SetLayerIndex(int I)	22
6.3.3.4	SetLayerName(string NewName)	22
6.3.3.5	ValidateReceive(Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)	22
6.3.3.6	ValidateSend(Workstation Destination, Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, int LayerIndex)	23
6.4	NSA.Model.NetworkComponents.Layers.DataLinkLayer Klassenreferenz	24
6.4.1	Ausführliche Beschreibung	25
6.4.2	Beschreibung der Konstruktoren und Destruktoren	25
6.4.2.1	DataLinkLayer(int I)	25
6.4.3	Dokumentation der Elementfunktionen	25
6.4.3.1	GetLayerIndex()	25
6.4.3.2	GetLayerName()	25
6.4.3.3	SetLayerIndex(int I)	25
6.4.3.4	SetLayerName(string NewName)	26
6.4.3.5	ValidateReceive(Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)	26
6.4.3.6	ValidateSend(Workstation Destination, Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, int LayerIndex)	26
6.5	NSA.Model.NetworkComponents.Hardwarenode Klassenreferenz	27
6.5.1	Ausführliche Beschreibung	29
6.5.2	Beschreibung der Konstruktoren und Destruktoren	29
6.5.2.1	Hardwarenode(string N)	29
6.5.3	Dokumentation der Elementfunktionen	29
6.5.3.1	AddConnection(string IfaceName, Connection Con)	29
6.5.3.2	AddInterface(IPAddress Ip, IPAddress Subnetmask, int PortNum=-1)	29
6.5.3.3	Equals(object Obj)	30

6.5.3.4	<code>Equals(Hardwarenode Other)</code>	31
6.5.3.5	<code>GetConnectionAtPort(string IfaceName)</code>	31
6.5.3.6	<code>GetHashCode()</code>	31
6.5.3.7	<code>GetInterfaceCount()</code>	31
6.5.3.8	<code>getNewInterfaceNumber()</code>	31
6.5.3.9	<code>GetPortIndexOfConnection(Connection C)</code>	31
6.5.3.10	<code>HasInterface(string IfaceName)</code>	32
6.5.3.11	<code>HasIp(IPAddress Ip)</code>	32
6.5.3.12	<code>InterfacelsUsed(string InterfaceName)</code>	33
6.5.3.13	<code>operator!=(Hardwarenode A, Hardwarenode B)</code>	33
6.5.3.14	<code>operator==(Hardwarenode A, Hardwarenode B)</code>	34
6.5.3.15	<code>Receive(Dictionary&lt; string, object &gt; Tags, ValidationInfo VallInfo, Hardwarenode Destination)</code>	34
6.5.3.16	<code>RemoveConnection(string IfaceName)</code>	34
6.5.3.17	<code>RemoveInterface(string InterfaceName)</code>	34
6.5.3.18	<code>Send(Hardwarenode Destination, Dictionary&lt; string, object &gt; Tags, ValidationInfo VallInfo)</code>	35
6.5.3.19	<code>SetInterface(string Ifacename, IPAddress Ip, IPAddress Mask)</code>	35
6.5.4	Dokumentation der Property's	35
6.5.4.1	Connections	35
6.5.4.2	Interfaces	35
6.5.4.3	Layerstack	36
6.5.4.4	Name	36
6.6	<code>NSA.Model.BusinessLogic.TestscenarioRunnables.HasInternetTestscenarioRunnable</code> Klassenreferenz	36
6.6.1	Beschreibung der Konstruktoren und Destruktoren	37
6.6.1.1	<code>HasInternetTestscenarioRunnable(Rule Rule)</code>	37
6.6.2	Dokumentation der Elementfunktionen	37
6.6.2.1	<code>Run()</code>	37
6.7	<code>NSA.Model.NetworkComponents.ILayer</code> Schnittstellenreferenz	38
6.7.1	Dokumentation der Elementfunktionen	39
6.7.1.1	<code>GetLayerIndex()</code>	39

6.7.1.2	GetLayerName()	39
6.7.1.3	SetLayerIndex(int I)	40
6.7.1.4	SetLayerName(string NewName)	40
6.7.1.5	ValidateReceive(Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)	41
6.7.1.6	ValidateSend(Workstation Destination, Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, int LayerIndex)	41
6.8	NSA.Model.NetworkComponents.Interface Klassenreferenz	42
6.8.1	Ausführliche Beschreibung	43
6.8.2	Beschreibung der Konstruktoren und Destruktoren	43
6.8.2.1	Interface(IPAddress Ip, IPAddress Mask, int Number)	43
6.8.3	Dokumentation der Elementfunktionen	43
6.8.3.1	SetInterface(IPAddress Ip, IPAddress Mask)	43
6.8.4	Dokumentation der Datenelemente	43
6.8.4.1	NamePrefix	43
6.8.5	Dokumentation der Property's	43
6.8.5.1	IpAddress	43
6.8.5.2	Name	44
6.8.5.3	Subnetmask	44
6.9	NSA.Model.NetworkComponents.Helper_Classes.IPAddressExtensions Klassenreferenz	44
6.9.1	Dokumentation der Elementfunktionen	44
6.9.1.1	GetBroadcastAddress(this IPAddress address, IPAddress subnetMask)	44
6.9.1.2	GetNetworkAddress(this IPAddress address, IPAddress subnetMask)	44
6.9.1.3	IsInSameSubnet(this IPAddress address2, IPAddress address, IPAddress subnetMask)	44
6.9.1.4	IsValidSubnetMask(this IPAddress Subnetmask)	44
6.10	NSA.Model.BusinessLogic.TestscenarioRunnables.ITestscenarioRunnable Schnittstellenreferenz	45
6.10.1	Dokumentation der Elementfunktionen	45
6.10.1.1	Run()	45
6.11	NSA.Model.NetworkComponents.Layerstack Klassenreferenz	45
6.11.1	Ausführliche Beschreibung	46
6.11.2	Beschreibung der Konstruktoren und Destruktoren	46

6.11.2.1	Layerstack()	46
6.11.3	Dokumentation der Elementfunktionen	46
6.11.3.1	AddLayer(ILayer Lay)	46
6.11.3.2	CreateUniqueName()	47
6.11.3.3	GetAllLayers()	47
6.11.3.4	GetLayer(int Index)	47
6.11.3.5	GetLayerByName(string Name)	48
6.11.3.6	GetSize()	48
6.11.3.7	InsertAt(int Index, ILayer Layer)	48
6.11.3.8	IsNameTaken(string Name)	48
6.11.3.9	RemoveLayer(string Name)	49
6.11.3.10	SetIndex(string Name, int NewIndex)	49
6.11.3.11	SetName(string OldName, string NewName)	50
6.12	NSA.Model.NetworkComponents.Network Klassenreferenz	51
6.12.1	Beschreibung der Konstruktoren und Destruktoren	51
6.12.1.1	Network()	51
6.12.2	Dokumentation der Elementfunktionen	51
6.12.2.1	AddConnection(string StartNodeInterfaceName, string EndNodeInterfaceName, Connection NewConnection)	51
6.12.2.2	AddHardwarenode(Hardwarenode NewNode)	52
6.12.2.3	GetAllConnections()	52
6.12.2.4	GetAllHardwarenodes()	52
6.12.2.5	GetAllWorkstations()	52
6.12.2.6	GetConnectionByName(string Name)	52
6.12.2.7	GetHardwarenodeByName(string Name)	53
6.12.2.8	GetHardwareNodesForSubnet(string Subnetmask)	53
6.12.2.9	GetRouters()	53
6.12.2.10	GetWorkstationByIp(IPAddress Ip)	53
6.12.2.11	RemoveConnection(string ConnectionName)	54
6.12.2.12	RemoveHardwarnode(string Name)	54
6.12.3	Dokumentation der Property's	54



6.12.3.1	Connections	54
6.13	NSA.Model.NetworkComponents.Layers.NetworkLayer Klassenreferenz	54
6.13.1	Ausführliche Beschreibung	56
6.13.2	Beschreibung der Konstruktoren und Destruktoren	56
6.13.2.1	NetworkLayer(int I)	56
6.13.3	Dokumentation der Elementfunktionen	56
6.13.3.1	GetLayerIndex()	56
6.13.3.2	GetLayerName()	56
6.13.3.3	SetLayerIndex(int I)	56
6.13.3.4	SetLayerName(string NewName)	57
6.13.3.5	ValidateReceive(Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)	57
6.13.3.6	ValidateSend(Workstation Destination, Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, int LayerIndex)	57
6.14	NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable Klassenreferenz	58
6.14.1	Beschreibung der Konstruktoren und Destruktoren	59
6.14.1.1	OnlyTestscenarioRunnable(Rule Rule, Network N)	59
6.14.2	Dokumentation der Elementfunktionen	59
6.14.2.1	Run()	59
6.15	NSA.Model.BusinessLogic.Packet Klassenreferenz	60
6.15.1	Ausführliche Beschreibung	60
6.15.2	Beschreibung der Konstruktoren und Destruktoren	60
6.15.2.1	Packet(Hardwarenode Src, Hardwarenode Dest, int T, bool ExpRes)	60
6.15.3	Dokumentation der Elementfunktionen	61
6.15.3.1	Send()	61
6.15.4	Dokumentation der Property's	61
6.15.4.1	Destination	61
6.15.4.2	ExpectedResult	62
6.15.4.3	Hops	62
6.15.4.4	Result	62
6.15.4.5	Source	62

6.15.4.6	Ttl	62
6.16	NSA.Model.NetworkComponents.Layers.PhysicalLayer Klassenreferenz	63
6.16.1	Ausführliche Beschreibung	64
6.16.2	Beschreibung der Konstruktoren und Destruktoren	64
6.16.2.1	PhysicalLayer(int I)	64
6.16.3	Dokumentation der Elementfunktionen	64
6.16.3.1	GetLayerIndex()	64
6.16.3.2	GetLayerName()	64
6.16.3.3	SetLayerIndex(int I)	64
6.16.3.4	SetLayerName(string NewName)	65
6.16.3.5	ValidateReceive(Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)	65
6.16.3.6	ValidateSend(Workstation Destination, Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, int LayerIndex)	65
6.17	NSA.Model.NetworkComponents.Layers.PresentationLayer Klassenreferenz	66
6.17.1	Ausführliche Beschreibung	67
6.17.2	Beschreibung der Konstruktoren und Destruktoren	67
6.17.2.1	PresentationLayer(int I)	67
6.17.3	Dokumentation der Elementfunktionen	67
6.17.3.1	GetLayerIndex()	67
6.17.3.2	GetLayerName()	68
6.17.3.3	SetLayerIndex(int I)	68
6.17.3.4	SetLayerName(string NewName)	68
6.17.3.5	ValidateReceive(Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)	68
6.17.3.6	ValidateSend(Workstation Destination, Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, int LayerIndex)	69
6.18	NSA.Model.BusinessLogic.Project Klassenreferenz	69
6.18.1	Ausführliche Beschreibung	69
6.18.2	Beschreibung der Konstruktoren und Destruktoren	70
6.18.2.1	Project()	70
6.18.3	Dokumentation der Property's	70

6.18.3.1	Network	70
6.18.3.2	Path	70
6.19	NSA.Model.NetworkComponents.Helper_Classes.Result Klassenreferenz	70
6.19.1	Ausführliche Beschreibung	71
6.19.2	Dokumentation der Aufzählungstypen	71
6.19.2.1	Errors	71
6.19.3	Beschreibung der Konstruktoren und Destruktoren	72
6.19.3.1	Result()	72
6.19.3.2	Result(Errors ErrId, string R, ILayer L)	72
6.19.4	Dokumentation der Datenelemente	72
6.19.4.1	ResultStrings	72
6.19.5	Dokumentation der Property's	73
6.19.5.1	ErrorId	73
6.19.5.2	LayerError	73
6.19.5.3	Res	73
6.19.5.4	SendError	73
6.20	NSA.Model.NetworkComponents.Route Klassenreferenz	73
6.20.1	Ausführliche Beschreibung	74
6.20.2	Beschreibung der Konstruktoren und Destruktoren	74
6.20.2.1	Route(IPAddress Destination, IPAddress Subnetmask, IPAddress Gateway, Interface Iface)	74
6.20.3	Dokumentation der Elementfunktionen	74
6.20.3.1	SetRoute(IPAddress DestinationIp, IPAddress Mask, IPAddress Gateway↔ Address, Interface Intface)	74
6.20.4	Dokumentation der Property's	75
6.20.4.1	Destination	75
6.20.4.2	Gateway	75
6.20.4.3	Iface	75
6.20.4.4	Name	75
6.20.4.5	Subnetmask	75
6.21	NSA.Model.NetworkComponents.Router Klassenreferenz	76

6.21.1 Ausführliche Beschreibung . . . . .	77
6.21.2 Beschreibung der Konstruktoren und Destruktoren . . . . .	77
6.21.2.1 Router(string Name) . . . . .	77
6.21.3 Dokumentation der Propertys . . . . .	77
6.21.3.1 IsGateway . . . . .	77
6.22 NSA.Model.BusinessLogic.Rule Klassenreferenz . . . . .	77
6.22.1 Ausführliche Beschreibung . . . . .	78
6.22.2 Beschreibung der Konstruktoren und Destruktoren . . . . .	78
6.22.2.1 Rule(string StartNode, List< string > EndNodes, Dictionary< string, int > Options, SimulationType SimulationType, bool ExpectedResult, Network N) . . . . .	78
6.22.3 Dokumentation der Elementfunktionen . . . . .	79
6.22.3.1 CheckForTrueOrFalse(string Text, string Rule) . . . . .	79
6.22.3.2 Parse(string Rule, Network N) . . . . .	79
6.22.4 Dokumentation der Datenelemente . . . . .	80
6.22.4.1 Parameters . . . . .	80
6.22.5 Dokumentation der Propertys . . . . .	80
6.22.5.1 EndNodes . . . . .	80
6.22.5.2 EndNodesString . . . . .	80
6.22.5.3 ExpectedResult . . . . .	80
6.22.5.4 Options . . . . .	80
6.22.5.5 SimulType . . . . .	80
6.22.5.6 StartNode . . . . .	80
6.22.5.7 StartNodeString . . . . .	81
6.23 NSA.Model.NetworkComponents.Layers.SessionLayer Klassenreferenz . . . . .	81
6.23.1 Ausführliche Beschreibung . . . . .	82
6.23.2 Beschreibung der Konstruktoren und Destruktoren . . . . .	82
6.23.2.1 SessionLayer(int I) . . . . .	82
6.23.3 Dokumentation der Elementfunktionen . . . . .	82
6.23.3.1 GetLayerIndex() . . . . .	82
6.23.3.2 GetLayerName() . . . . .	83
6.23.3.3 SetLayerIndex(int I) . . . . .	83

6.23.3.4	SetLayerName(string NewName)	83
6.23.3.5	ValidateReceive(Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)	83
6.23.3.6	ValidateSend(Workstation Destination, Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, int LayerIndex)	84
6.24	NSA.Model.BusinessLogic.TestscenarioRunnables.SimpleTestscenarioRunnable Klassenreferenz	84
6.24.1	Beschreibung der Konstruktoren und Destruktoren	85
6.24.1.1	SimpleTestscenarioRunnable(Rule Rule)	85
6.24.2	Dokumentation der Elementfunktionen	85
6.24.2.1	Run()	85
6.25	NSA.Model.BusinessLogic.Simulation Klassenreferenz	86
6.25.1	Ausführliche Beschreibung	86
6.25.2	Beschreibung der Konstruktoren und Destruktoren	86
6.25.2.1	Simulation(string I)	86
6.25.2.2	Simulation(string I, string S, string D, bool ExpRes)	87
6.25.3	Dokumentation der Elementfunktionen	87
6.25.3.1	AddPacketSend(Packet Packet)	87
6.25.3.2	Execute()	87
6.25.3.3	GetAllPackets()	87
6.25.3.4	GetLastPacket()	88
6.25.4	Dokumentation der Property's	88
6.25.4.1	Destination	88
6.25.4.2	ExpectedResult	88
6.25.4.3	Id	88
6.25.4.4	PacketsReceived	88
6.25.4.5	PacketsSend	88
6.25.4.6	Source	88
6.26	NSA.Model.NetworkComponents.Switch Klassenreferenz	89
6.26.1	Ausführliche Beschreibung	90
6.26.2	Beschreibung der Konstruktoren und Destruktoren	90
6.26.2.1	Switch(string Name)	90

6.26.3	Dokumentation der Elementfunktionen	90
6.26.3.1	AddInterface(IPAddress Ip, IPAddress Subnetmask, int PortNum=-1)	90
6.26.3.2	Send(Hardwarenode Destination, Dictionary< string, object > Tags, ValidationInfo VallInfo)	90
6.26.3.3	SendToDestination(Workstation Destination, ValidationInfo VallInfo, Connection ComingCon, IPAddress nodeIP, IPAddress subnetmask)	91
6.26.3.4	SendToIp(ValidationInfo VallInfo, Connection ComingConn)	92
6.26.3.5	SetInterfaceCount(int Count)	93
6.27	NSA.Model.BusinessLogic.Testscenario Klassenreferenz	94
6.27.1	Beschreibung der Konstruktoren und Destruktoren	94
6.27.1.1	Testscenario(string T, Network N, string FileName)	94
6.27.2	Dokumentation der Elementfunktionen	94
6.27.2.1	GetTestscenarioRunnables()	94
6.27.3	Dokumentation der Property's	95
6.27.3.1	FileName	95
6.27.3.2	Id	95
6.28	NSA.Model.NetworkComponents.Layers.TransportLayer Klassenreferenz	95
6.28.1	Ausführliche Beschreibung	96
6.28.2	Beschreibung der Konstruktoren und Destruktoren	96
6.28.2.1	TransportLayer(int I)	96
6.28.3	Dokumentation der Elementfunktionen	96
6.28.3.1	GetLayerIndex()	96
6.28.3.2	GetLayerName()	97
6.28.3.3	SetLayerIndex(int I)	97
6.28.3.4	SetLayerName(string NewName)	97
6.28.3.5	ValidateReceive(Workstation CurrentNode, ValidationInfo VallInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)	97
6.28.3.6	ValidateSend(Workstation Destination, Workstation CurrentNode, ValidationInfo VallInfo, Dictionary< string, object > Tags, int LayerIndex)	98
6.29	NSA.Model.NetworkComponents.Helper_Classes.ValidationInfo Klassenreferenz	98
6.29.1	Ausführliche Beschreibung	98
6.29.2	Dokumentation der Property's	99

6.29.2.1	<code>Iface</code> . . . . .	99
6.29.2.2	<code>NextNodeIp</code> . . . . .	99
6.29.2.3	<code>NextNodes</code> . . . . .	99
6.29.2.4	<code>Res</code> . . . . .	99
6.29.2.5	<code>Source</code> . . . . .	99
6.30	<code>NSA.Model.NetworkComponents.Workstation</code> Klassenreferenz . . . . .	100
6.30.1	Ausführliche Beschreibung . . . . .	101
6.30.2	Beschreibung der Konstruktoren und Destrukturen . . . . .	101
6.30.2.1	<code>Workstation(string Name)</code> . . . . .	101
6.30.3	Dokumentation der Elementfunktionen . . . . .	102
6.30.3.1	<code>AddRoute(Route Route)</code> . . . . .	102
6.30.3.2	<code>GetRouteAt(int Index)</code> . . . . .	102
6.30.3.3	<code>GetRouteCount()</code> . . . . .	102
6.30.3.4	<code>GetRoutes()</code> . . . . .	102
6.30.3.5	<code>HasIp(IPAddress Ip)</code> . . . . .	103
6.30.3.6	<code>Receive(Dictionary&lt; string, object &gt; Tags, ValidationInfo VallInfo, Hardwarenode Destination)</code> . . . . .	103
6.30.3.7	<code>RemoveRoute(string N)</code> . . . . .	103
6.30.3.8	<code>Send(Hardwarenode Destination, Dictionary&lt; string, object &gt; Tags, ValidationInfo VallInfo)</code> . . . . .	104
6.30.3.9	<code>SetRoute(string RouteName, IPAddress Destination, IPAddress Subnetmask, IPAddress Gateway, Interface Iface)</code> . . . . .	104
6.30.4	Dokumentation der Property's . . . . .	105
6.30.4.1	<code>StandardGateway</code> . . . . .	105
6.30.4.2	<code>StandardGatewayPort</code> . . . . .	105

<b>7</b>	<b>Datei-Dokumentation</b>	<b>107</b>
7.1	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ BusinessLogic/Packet.cs-Dateireferenz . . . . .	107
7.2	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ BusinessLogic/Project.cs-Dateireferenz . . . . .	107
7.3	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ BusinessLogic/Rule.cs-Dateireferenz . . . . .	107
7.4	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ BusinessLogic/Simulation.cs-Dateireferenz . . . . .	108
7.5	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ BusinessLogic/Testscenario.cs-Dateireferenz . . . . .	108
7.6	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ BusinessLogic/TestscenarioRunnables/HasInternetTestscenarioRunnable.cs-Dateireferenz . . . . .	108
7.7	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ BusinessLogic/TestscenarioRunnables/ITestscenarioRunnable.cs-Dateireferenz . . . . .	109
7.8	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ BusinessLogic/TestscenarioRunnables/OnlyTestscenarioRunnable.cs-Dateireferenz . . . . .	109
7.9	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ BusinessLogic/TestscenarioRunnables/SimpleTestscenarioRunnable.cs-Dateireferenz . . . . .	109
7.10	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Connection.cs-Dateireferenz . . . . .	110
7.11	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/HardwareNode.cs-Dateireferenz . . . . .	110
7.12	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Helper Classes/IPAddressExtensions.cs-Dateireferenz . . . . .	110
7.13	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Helper Classes/Result.cs-Dateireferenz . . . . .	110
7.14	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Helper Classes/ValidationInfo.cs-Dateireferenz . . . . .	111
7.15	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/ILayer.cs-Dateireferenz . . . . .	111
7.16	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Interface.cs-Dateireferenz . . . . .	111
7.17	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Layers/ApplicationLayer.cs-Dateireferenz . . . . .	112
7.18	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Layers/CustomLayer.cs-Dateireferenz . . . . .	112
7.19	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Layers/DataLinkLayer.cs-Dateireferenz . . . . .	112



7.20	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Layers/NetworkLayer.cs-Dateireferenz . . . . .	112
7.21	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Layers/PhysicalLayer.cs-Dateireferenz . . . . .	113
7.22	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Layers/PresentationLayer.cs-Dateireferenz . . . . .	113
7.23	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Layers/SessionLayer.cs-Dateireferenz . . . . .	113
7.24	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Layers/TransportLayer.cs-Dateireferenz . . . . .	114
7.25	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Layerstack.cs-Dateireferenz . . . . .	114
7.26	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Network.cs-Dateireferenz . . . . .	114
7.27	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Route.cs-Dateireferenz . . . . .	114
7.28	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Router.cs-Dateireferenz . . . . .	115
7.29	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Switch.cs-Dateireferenz . . . . .	115
7.30	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ NetworkComponents/Workstation.cs-Dateireferenz . . . . .	115
7.31	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Debug/TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs-Dateireferenz . . . . .	116
7.32	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Release/TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs-Dateireferenz . . . . .	116
7.33	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Debug/TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dda1.cs-Dateireferenz . . . . .	116
7.34	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Release/TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dda1.cs-Dateireferenz . . . . .	116
7.35	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Debug/TemporaryGeneratedFile_E7A71F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs-Dateireferenz . . . . .	116
7.36	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Release/TemporaryGeneratedFile_E7A71F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs-Dateireferenz . . . . .	116
7.37	C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/↔ Properties/AssemblyInfo.cs-Dateireferenz . . . . .	116



# 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.BusinessLogic.TestscenarioRunnables</a> . . . . .	10
<a href="#">NSA.Model.NetworkComponents</a> . . . . .	10
<a href="#">NSA.Model.NetworkComponents.Helper_Classes</a> . . . . .	11
<a href="#">NSA.Model.NetworkComponents.Layers</a> . . . . .	11



## Kapitel 2

# Hierarchie-Verzeichnis

### 2.1 Klassenhierarchie

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

NSA.Model.NetworkComponents.Connection . . . . .	16
NSA.Model.NetworkComponents.Hardwarenode . . . . .	27
NSA.Model.NetworkComponents.Switch . . . . .	89
NSA.Model.NetworkComponents.Workstation . . . . .	100
NSA.Model.NetworkComponents.Router . . . . .	76
NSA.Model.NetworkComponents.ILayer . . . . .	38
NSA.Model.NetworkComponents.Layers.ApplicationLayer . . . . .	13
NSA.Model.NetworkComponents.Layers.CustomLayer . . . . .	20
NSA.Model.NetworkComponents.Layers.DataLinkLayer . . . . .	24
NSA.Model.NetworkComponents.Layers.NetworkLayer . . . . .	54
NSA.Model.NetworkComponents.Layers.PhysicalLayer . . . . .	63
NSA.Model.NetworkComponents.Layers.PresentationLayer . . . . .	66
NSA.Model.NetworkComponents.Layers.SessionLayer . . . . .	81
NSA.Model.NetworkComponents.Layers.TransportLayer . . . . .	95
NSA.Model.NetworkComponents.Interface . . . . .	42
NSA.Model.NetworkComponents.Helper_Classes.IPAddressExtensions . . . . .	44
NSA.Model.BusinessLogic.TestscenarioRunnables.ITestscenarioRunnable . . . . .	45
NSA.Model.BusinessLogic.TestscenarioRunnables.HasInternetTestscenarioRunnable . . . . .	36
NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable . . . . .	58
NSA.Model.BusinessLogic.TestscenarioRunnables.SimpleTestscenarioRunnable . . . . .	84
NSA.Model.NetworkComponents.Layerstack . . . . .	45
NSA.Model.NetworkComponents.Network . . . . .	51
NSA.Model.BusinessLogic.Packet . . . . .	60
NSA.Model.BusinessLogic.Project . . . . .	69
NSA.Model.NetworkComponents.Helper_Classes.Result . . . . .	70
NSA.Model.NetworkComponents.Route . . . . .	73
NSA.Model.BusinessLogic.Rule . . . . .	77
NSA.Model.BusinessLogic.Simulation . . . . .	86
NSA.Model.BusinessLogic.Testscenario . . . . .	94
NSA.Model.NetworkComponents.Helper_Classes.ValidationInfo . . . . .	98



## Kapitel 3

# Klassen-Verzeichnis

### 3.1 Auflistung der Klassen

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

<a href="#">NSA.Model.NetworkComponents.Layers.ApplicationLayer</a>	
Application-Layer . . . . .	13
<a href="#">NSA.Model.NetworkComponents.Connection</a>	
Class for a connection between two hardwarenodes. . . . .	16
<a href="#">NSA.Model.NetworkComponents.Layers.CustomLayer</a>	
Custom-Layer . . . . .	20
<a href="#">NSA.Model.NetworkComponents.Layers.DataLinkLayer</a>	
DataLink-Layer . . . . .	24
<a href="#">NSA.Model.NetworkComponents.Hardwarenode</a>	
Implements the basis class for hardwarenodes. . . . .	27
<a href="#">NSA.Model.BusinessLogic.TestscenarioRunnables.HasInternetTestscenarioRunnable</a>	36
<a href="#">NSA.Model.NetworkComponents.ILayer</a>	38
<a href="#">NSA.Model.NetworkComponents.Interface</a>	
Class for a interface of an hardwarenode. . . . .	42
<a href="#">NSA.Model.NetworkComponents.Helper_Classes.IPAddressExtensions</a>	44
<a href="#">NSA.Model.BusinessLogic.TestscenarioRunnables.ITestscenarioRunnable</a>	45
<a href="#">NSA.Model.NetworkComponents.Layerstack</a>	
Layerstack . . . . .	45
<a href="#">NSA.Model.NetworkComponents.Network</a>	51
<a href="#">NSA.Model.NetworkComponents.Layers.NetworkLayer</a>	
Network-Layer . . . . .	54
<a href="#">NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable</a>	58
<a href="#">NSA.Model.BusinessLogic.Packet</a>	
Class for a packet . . . . .	60
<a href="#">NSA.Model.NetworkComponents.Layers.PhysicalLayer</a>	
Physical-Layer . . . . .	63
<a href="#">NSA.Model.NetworkComponents.Layers.PresentationLayer</a>	
Presentation-Layer . . . . .	66
<a href="#">NSA.Model.BusinessLogic.Project</a>	
Class for project . . . . .	69
<a href="#">NSA.Model.NetworkComponents.Helper_Classes.Result</a>	
Result class for the packetresult . . . . .	70
<a href="#">NSA.Model.NetworkComponents.Route</a>	
class for a single route of the routingtable . . . . .	73
<a href="#">NSA.Model.NetworkComponents.Router</a>	
Implements the network component <a href="#">Router</a> . . . . .	76

<a href="#">NSA.Model.BusinessLogic.Rule</a>	
Rule class parses the text and if it is valid, the <a href="#">Rule</a> object will be used for simulations	77
<a href="#">NSA.Model.NetworkComponents.Layers.SessionLayer</a>	
Session-Layer	81
<a href="#">NSA.Model.BusinessLogic.TestscenarioRunnables.SimpleTestscenarioRunnable</a>	84
<a href="#">NSA.Model.BusinessLogic.Simulation</a>	
Class for the simulation	86
<a href="#">NSA.Model.NetworkComponents.Switch</a>	
Implements the network component switch.	89
<a href="#">NSA.Model.BusinessLogic.Testscenario</a>	94
<a href="#">NSA.Model.NetworkComponents.Layers.TransportLayer</a>	
Transport-Layer	95
<a href="#">NSA.Model.NetworkComponents.Helper_Classes.ValidationInfo</a>	
Helper class for the parameters of the simulation	98
<a href="#">NSA.Model.NetworkComponents.Workstation</a>	
Implements the network component <a href="#">Workstation</a> .	100



## 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> . . . . .	107
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↔ Logic/ <a href="#">Project.cs</a> . . . . .	107
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↔ Logic/ <a href="#">Rule.cs</a> . . . . .	107
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↔ Logic/ <a href="#">Simulation.cs</a> . . . . .	108
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↔ Logic/ <a href="#">Testscenario.cs</a> . . . . .	108
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↔ Logic/TestscenarioRunnables/ <a href="#">HasInternetTestscenarioRunnable.cs</a> . . . . .	108
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↔ Logic/TestscenarioRunnables/ <a href="#">ITestscenarioRunnable.cs</a> . . . . .	109
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↔ Logic/TestscenarioRunnables/ <a href="#">OnlyTestscenarioRunnable.cs</a> . . . . .	109
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Business↔ Logic/TestscenarioRunnables/ <a href="#">SimpleTestscenarioRunnable.cs</a> . . . . .	109
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Connection.cs</a> . . . . .	110
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Hardwarenode.cs</a> . . . . .	110
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">ILayer.cs</a> . . . . .	111
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Interface.cs</a> . . . . .	111
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Layerstack.cs</a> . . . . .	114
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Network.cs</a> . . . . .	114
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Route.cs</a> . . . . .	114
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Router.cs</a> . . . . .	115
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Switch.cs</a> . . . . .	115

C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/ <a href="#">Workstation.cs</a> . . . . .	115
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Helper Classes/ <a href="#">IPAddressExtensions.cs</a> . . . . .	110
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Helper Classes/ <a href="#">Result.cs</a> . . . . .	110
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Helper Classes/ <a href="#">ValidationInfo.cs</a> . . . . .	111
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Layers/ <a href="#">ApplicationLayer.cs</a> . . . . .	112
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Layers/ <a href="#">CustomLayer.cs</a> . . . . .	112
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Layers/ <a href="#">DataLinkLayer.cs</a> . . . . .	112
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Layers/ <a href="#">NetworkLayer.cs</a> . . . . .	112
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Layers/ <a href="#">PhysicalLayer.cs</a> . . . . .	113
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Layers/ <a href="#">PresentationLayer.cs</a> . . . . .	113
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Layers/ <a href="#">SessionLayer.cs</a> . . . . .	113
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Network↔ Components/Layers/ <a href="#">TransportLayer.cs</a> . . . . .	114
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Debug/ <a href="#">TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs</a> . . . . .	116
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Debug/ <a href="#">TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dda1.cs</a> . . . . .	116
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Debug/ <a href="#">TemporaryGeneratedFile_E7A71F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs</a> . . . . .	116
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Release/ <a href="#">TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs</a> . . . . .	116
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Release/ <a href="#">TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dda1.cs</a> . . . . .	116
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/obj/↔ Release/ <a href="#">TemporaryGeneratedFile_E7A71F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs</a> . . . . .	116
C:/SWP16/Basisverzeichnis/trunk/03_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/Properties/ <a href="#">Assembly↔ Info.cs</a> . . . . .	116

# Kapitel 5

## Dokumentation der Namensbereiche

### 5.1 NSA-Namensbereichsreferenz

#### Namensbereiche

- namespace [Model](#)

### 5.2 NSA.Model-Namensbereichsreferenz

#### Namensbereiche

- namespace [BusinessLogic](#)
- namespace [NetworkComponents](#)

### 5.3 NSA.Model.BusinessLogic-Namensbereichsreferenz

#### Namensbereiche

- namespace [TestscenarioRunnables](#)

#### Klassen

- class [Packet](#)  
*Class for a packet*
- class [Project](#)  
*Class for project*
- class [Rule](#)  
*[Rule](#) class parses the text and if it is valid, the [Rule](#) object will be used for simulations*
- class [Simulation](#)  
*Class for the simulation*
- class [Testscenario](#)

## Aufzählungen

- enum [SimulationType](#) { [SimulationType.Simple](#) = 0, [SimulationType.Only](#) = 1, [SimulationType.HasInternet](#) = 2 }

*Simulation types*

### 5.3.1 Dokumentation der Aufzählungstypen

#### 5.3.1.1 enum [NSA.Model.BusinessLogic.SimulationType](#) [strong]

[Simulation](#) types

Aufzählungswerte

***Simple***  
***Only***  
***HasInternet***

## 5.4 NSA.Model.BusinessLogic.TestscenarioRunnables-Namensbereichsreferenz

### Klassen

- class [HasInternetTestscenarioRunnable](#)
- interface [ITestscenarioRunnable](#)
- class [OnlyTestscenarioRunnable](#)
- class [SimpleTestscenarioRunnable](#)

## 5.5 NSA.Model.NetworkComponents-Namensbereichsreferenz

### Namensbereiche

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

### Klassen

- class [Connection](#)  
*Class for a connection between two hardwarenodes.*
- class [Hardwarenode](#)  
*Implements the basis class for hardwarenodes.*
- interface [ILayer](#)
- class [Interface](#)  
*Class for a interface of an hardwarenode.*
- class [Layerstack](#)  
[Layerstack](#)
- class [Network](#)
- class [Route](#)  
*class for a single route of the routingtable*
- class [Router](#)  
*Implements the network component [Router](#).*
- class [Switch](#)  
*Implements the network component switch.*
- class [Workstation](#)  
*Implements the network component [Workstation](#).*

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

### Klassen

- class [IPAddressExtensions](#)
- class [Result](#)  
*Result class for the packetresult*
- class [ValidationInfo](#)  
*Helper class for the parameters of the simulation*

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

### Klassen

- class [ApplicationLayer](#)  
*Application-Layer*
- class [CustomLayer](#)  
*Custom-Layer*
- class [DataLinkLayer](#)  
*DataLink-Layer*
- class [NetworkLayer](#)  
*Network-Layer*
- class [PhysicalLayer](#)  
*Physical-Layer*
- class [PresentationLayer](#)  
*Presentation-Layer*
- class [SessionLayer](#)  
*Session-Layer*
- class [TransportLayer](#)  
*Transport-Layer*



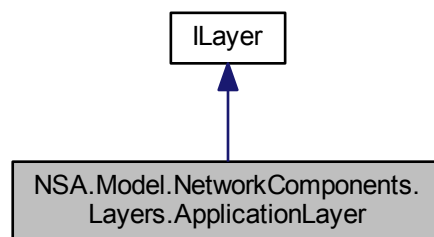
## Kapitel 6

# Klassen-Dokumentation

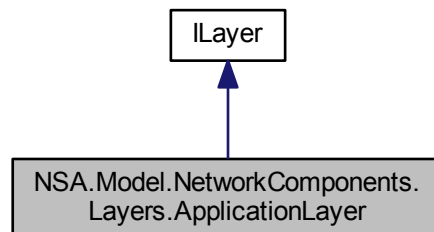
### 6.1 NSA.Model.NetworkComponents.Layers.ApplicationLayer Klassenreferenz

Application-Layer

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



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



## Öffentliche Methoden

- [ApplicationLayer](#) (int I)  
*Initializes a new instance of the [ApplicationLayer](#) class.*
- bool [ValidateReceive](#) ([Workstation](#) CurrentNode, [ValidationInfo](#) ValInfo, Dictionary< string, object > Tags, [Hardwarenode](#) Destination, int LayerIndex)  
*Validates the layer while receiving a packet.*
- string [GetLayerName](#) ()  
*Gets the name of the layer.*
- bool [SetLayerName](#) (string NewName)  
*Sets the name of the layer.*
- int [GetLayerIndex](#) ()  
*Gets the index of the layer.*
- void [SetLayerIndex](#) (int I)  
*Sets the index of the layer.*
- void [ValidateSend](#) ([Workstation](#) Destination, [Workstation](#) CurrentNode, [ValidationInfo](#) ValInfo, Dictionary< string, object > Tags, int LayerIndex)  
*Validates the layer while sending a packet.*

### 6.1.1 Ausführliche Beschreibung

Application-Layer

Siehe auch

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

### 6.1.2 Beschreibung der Konstruktoren und Destruktoren

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

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

Parameter

I	The index.
---	------------

### 6.1.3 Dokumentation der Elementfunktionen

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

Gets the index of the layer.

Rückgabe

The index

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



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

Gets the name of the layer.

Rückgabe

The Layername

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

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

Sets the index of the layer.

Parameter

I	The Index.
---	------------

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

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

Sets the name of the layer.

Parameter

<i>NewName</i>	The new name
----------------	--------------

Rückgabe

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

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

Validates the layer while receiving a packet.

Parameter

<i>CurrentNode</i>	Current node
<i>ValInfo</i>	Validation Info
<i>Tags</i>	Tags
<i>Destination</i>	Destinationnode
<i>LayerIndex</i>	Index of the Layer

**Rückgabe**

Boolean value indicating if the validation was successfull

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

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

Validates the layer while sending a packet.

**Parameter**

<i>Destination</i>	The Destination
<i>CurrentNode</i>	Current Node
<i>VallInfo</i>	Validation Info
<i>Tags</i>	Tags
<i>LayerIndex</i>	The Layer index

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

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

## 6.2 [NSA.Model.NetworkComponents.Connection](#) Klassenreferenz

Class for a connection between two hardwarenodes.

**Öffentliche Methoden**

- [Connection](#) ([Hardwarenode](#) Source, [Hardwarenode](#) Target)  
*Initializes a new instance of the [Connection](#) class.*
- override bool [Equals](#) (object Obj)  
*Determines whether the specified System.Object, is equal to this instance.*
- bool [Equals](#) ([Connection](#) Other)  
*Equalses the specified other.*
- override int [GetHashCode](#) ()  
*Returns a hash code for this instance.*
- int [GetPortIndex](#) ([Hardwarenode](#) Node)  
*Gets the index of the port.*

**Öffentliche, statische Methoden**

- static bool [operator==](#) ([Connection](#) A, [Connection](#) B)  
*Implements the operator ==.*
- static bool [operator!=](#) ([Connection](#) A, [Connection](#) B)  
*Implements the operator !=.*

## Propertys

- [Hardwarenode Start](#) [get]  
*Gets the start hardwarenode.*
- [Hardwarenode End](#) [get]  
*Gets the end hardwarenode.*
- string [Name](#) [get]  
*Gets the name (id) of the connection. Every connection has a unique id.*

### 6.2.1 Ausführliche Beschreibung

Class for a connection between two hardwarenodes.

### 6.2.2 Beschreibung der Konstruktoren und Destruktoren

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

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

##### Parameter

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

### 6.2.3 Dokumentation der Elementfunktionen

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

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

##### Parameter

<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, wo diese Funktion aufgerufen wird:



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

Equalses the specified other.

#### Parameter

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

#### Rückgabe

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

Returns a hash code for this instance.

#### Rückgabe

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

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

Gets the index of the port.

#### Parameter

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

#### Rückgabe

Portindex

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



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

Implements the operator !=.

**Parameter**

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

**Rückgabe**

The result of the operator.

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

Implements the operator ==.

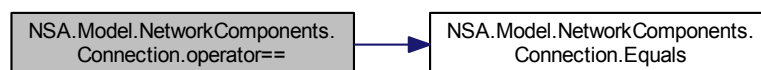
**Parameter**

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

**Rückgabe**

The result of the operator.

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



## 6.2.4 Dokumentation der Propertys

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

Gets the end hardwarenode.

The end.

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

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

The name.

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

Gets the start hardwarenode.

The start.

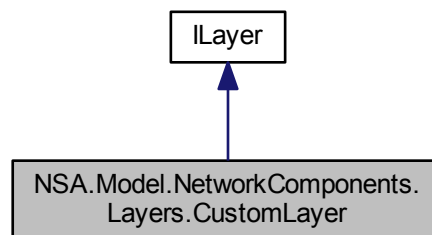
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

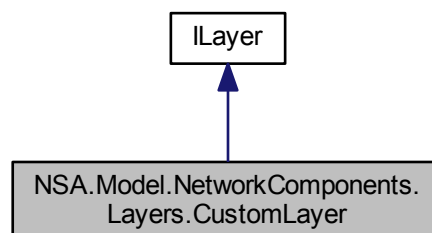
### 6.3 NSA.Model.NetworkComponents.Layers.CustomLayer Klassenreferenz

Custom-Layer

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



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



## Öffentliche Methoden

- [CustomLayer](#) (string N, int I)  
*Initializes a new instance of the [CustomLayer](#) class.*
- void [ValidateSend](#) ([Workstation](#) Destination, [Workstation](#) CurrentNode, [ValidationInfo](#) ValInfo, Dictionary< string, object > Tags, int LayerIndex)  
*Validates the layer while sending a packet.*
- bool [ValidateReceive](#) ([Workstation](#) CurrentNode, [ValidationInfo](#) ValInfo, Dictionary< string, object > Tags, [Hardwarenode](#) Destination, int LayerIndex)  
*Validates the layer while receiving a packet.*
- string [GetLayerName](#) ()  
*Gets the name of the layer.*
- bool [SetLayerName](#) (string NewName)  
*Sets the name of the layer.*
- int [GetLayerIndex](#) ()  
*Gets the index of the layer.*
- void [SetLayerIndex](#) (int I)  
*Sets the index of the layer.*

### 6.3.1 Ausführliche Beschreibung

Custom-Layer

Siehe auch

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

### 6.3.2 Beschreibung der Konstruktoren und Destruktoren

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

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

Parameter

N	The name.
I	The index.

### 6.3.3 Dokumentation der Elementfunktionen

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

Gets the index of the layer.

Rückgabe

The index

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

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

Gets the name of the layer.

#### Rückgabe

The Layername

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

### 6.3.3.3 void NSA.Model.NetworkComponents.Layers.CustomLayer.SetLayerIndex ( int / )

Sets the index of the layer.

#### Parameter

/	The Index.
---	------------

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

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

Sets the name of the layer.

#### Parameter

<i>NewName</i>	The new Name
----------------	--------------

#### Rückgabe

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

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

Validates the layer while receiving a packet.

#### Parameter

<i>CurrentNode</i>	
<i>ValInfo</i>	
<i>Tags</i>	
<i>Destination</i>	
<i>LayerIndex</i>	

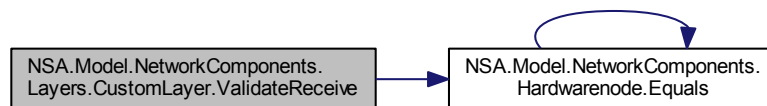


**Rückgabe**

Boolean value indicating if the validation was successfull

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

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



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

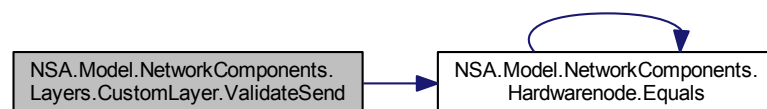
Validates the layer while sending a packet.

**Parameter**

<i>Destination</i>	The Destination
<i>CurrentNode</i>	Current Node
<i>VallInfo</i>	Validation Info
<i>Tags</i>	Tags
<i>LayerIndex</i>	The Layer index

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

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



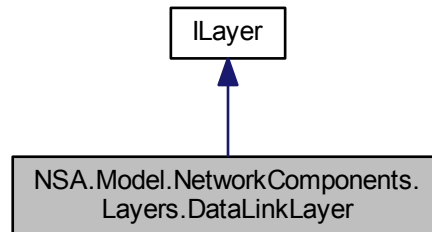
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

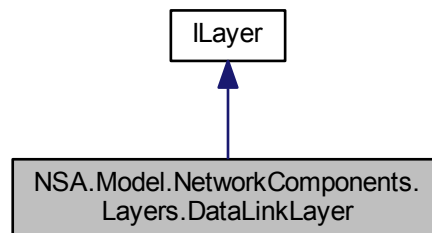
## 6.4 NSA.Model.NetworkComponents.Layers.DataLinkLayer Klassenreferenz

DataLink-Layer

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



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



### Öffentliche Methoden

- `DataLinkLayer` (int I)  
*Initializes a new instance of the `DataLinkLayer` class.*
- void `ValidateSend` (`Workstation` Destination, `Workstation` CurrentNode, `ValidationInfo` ValInfo, Dictionary< string, object > Tags, int LayerIndex)  
*Validates the layer while sending a packet.*
- bool `ValidateReceive` (`Workstation` CurrentNode, `ValidationInfo` ValInfo, Dictionary< string, object > Tags, `Hardwarenode` Destination, int LayerIndex)  
*Validates the layer while receiving a packet.*
- string `GetLayerName` ()  
*Gets the name of the layer.*
- bool `SetLayerName` (string NewName)  
*Sets the name of the layer.*

- int [GetLayerIndex](#) ()  
*Gets the index of the layer.*
- void [SetLayerIndex](#) (int I)  
*Sets the index of the layer.*

### 6.4.1 Ausführliche Beschreibung

DataLink-Layer

Siehe auch

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

### 6.4.2 Beschreibung der Konstruktoren und Destruktoren

#### 6.4.2.1 NSA.Model.NetworkComponents.Layers.DataLinkLayer.DataLinkLayer ( int I )

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

Parameter

/	The index.
---	------------

### 6.4.3 Dokumentation der Elementfunktionen

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

Gets the index of the layer.

Rückgabe

The index

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

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

Gets the name of the layer.

Rückgabe

The Layername

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

#### 6.4.3.3 void NSA.Model.NetworkComponents.Layers.DataLinkLayer.SetLayerIndex ( int I )

Sets the index of the layer.

## Parameter

/	The Index.
---	------------

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

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

Sets the name of the layer.

## Parameter

<i>NewName</i>	New Name
----------------	----------

## Rückgabe

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

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

Validates the layer while receiving a packet.

## Parameter

<i>CurrentNode</i>	Current node
<i>VallInfo</i>	Validation Info
<i>Tags</i>	Tags
<i>Destination</i>	Destinationnode
<i>LayerIndex</i>	Index of the Layer

## Rückgabe

Boolean value indicating if the validation was successfull

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

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

Validates the layer while sending a packet.

## Parameter

<i>Destination</i>	The Destination
--------------------	-----------------

## Parameter

<i>CurrentNode</i>	Current Node
<i>VallInfo</i>	Validation Info
<i>Tags</i>	Tags
<i>LayerIndex</i>	The Layer index

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

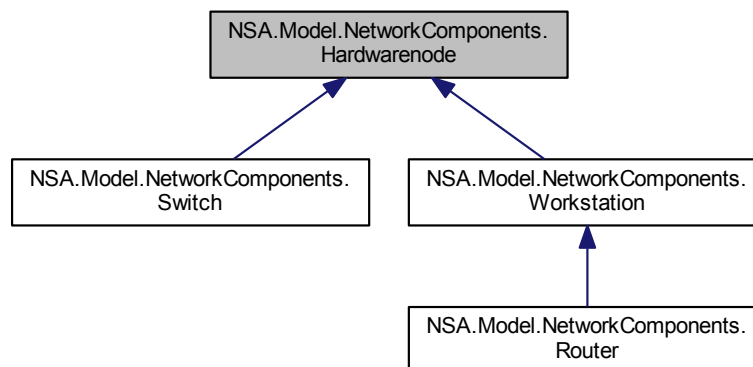
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

## 6.5 NSA.Model.NetworkComponents.Hardwarenode Klassenreferenz

Implements the basis class for hardwarenodes.

Klassendiagramm für NSA.Model.NetworkComponents.Hardwarenode:



### Öffentliche Methoden

- [Hardwarenode](#) (string N)  
*Initializes a new instance of the [Hardwarenode](#) class.*
- virtual [Interface AddInterface](#) (IPAddress Ip, IPAddress Subnetmask, int PortNum=-1)  
*Adds a new interface with the given IP and subnetmask*
- void [RemoveInterface](#) (string InterfaceName)  
*Removes the interface with the given name.*
- int [GetInterfaceCount](#) ()  
*Gets the interface count.*
- virtual void [SetInterface](#) (string Ifacename, IPAddress Ip, IPAddress Mask)  
*Sets the interface.*

- virtual bool [HasInterface](#) (string lfaceName)  
*Determines if there is an [Interface](#) with the specified name.*
- void [AddConnection](#) (string lfaceName, [Connection](#) Con)  
*Adds a connection.*
- virtual void [RemoveConnection](#) (string lfaceName)  
*Removes a connection.*
- [Connection](#) [GetConnectionAtPort](#) (string lfaceName)  
*Gets the connection at port.*
- bool [InterfacesUsed](#) (string InterfaceName)  
*Interfaces the is used.*
- virtual bool [HasIp](#) (IPAddress Ip)  
*Checks if the [Hardwarenode](#) has the IP*
- virtual List< [Hardwarenode](#) > [Send](#) ([Hardwarenode](#) Destination, Dictionary< string, object > Tags, [ValidationInfo](#) VallInfo)  
*[Hardwarenode](#) sends the package to specified destination.*
- virtual bool [Receive](#) (Dictionary< string, object > Tags, [ValidationInfo](#) VallInfo, [Hardwarenode](#) Destination)  
*[Hardwarenode](#) receives the package.*
- override bool [Equals](#) (object Obj)  
*Determines whether the specified System.Object, is equal to this instance.*
- bool [Equals](#) ([Hardwarenode](#) Other)  
*Equalses the specified other.*
- override int [GetHashCode](#) ()  
*Returns a hash code for this instance.*
- int [GetPortIndexOfConnection](#) ([Connection](#) C)  
*Gets the port index of connection.*

## Öffentliche, statische Methoden

- static bool [operator==](#) ([Hardwarenode](#) A, [Hardwarenode](#) B)  
*Implements the operator ==.*
- static bool [operator!=](#) ([Hardwarenode](#) A, [Hardwarenode](#) B)  
*Implements the operator !=.*

## Geschützte Methoden

- int [getNewInterfaceNumber](#) ()  
*Gets the new interface number.*

## Property

- [Layerstack](#) [Layerstack](#) [get]  
*Gets the layerstack.*
- Dictionary< string, [Connection](#) > [Connections](#) = new [Layerstack](#)() [get, protected set]  
*Gets or sets the connections.*
- string [Name](#) = new Dictionary<string, [Connection](#)>() [get, set]  
*Gets or sets the name of the hardwarenode.*
- List< [Interface](#) > [Interfaces](#) [get, protected set]  
*Gets or sets the interfaces.*

### 6.5.1 Ausführliche Beschreibung

Implements the basis class for hardwarenodes.

### 6.5.2 Beschreibung der Konstruktoren und Destruktoren

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

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

Parameter

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

### 6.5.3 Dokumentation der Elementfunktionen

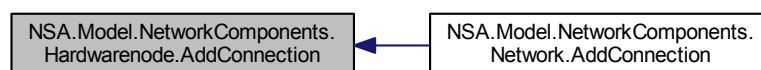
#### 6.5.3.1 void NSA.Model.NetworkComponents.Hardwarenode.AddConnection ( string *IfaceName*, Connection *Con* )

Adds a connection.

Parameter

<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:



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

Adds a new interface with the given IP and subnetmask

Parameter

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

### Rückgabe

The newly added [Interface](#)

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

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

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

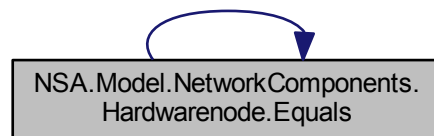
#### Parameter

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

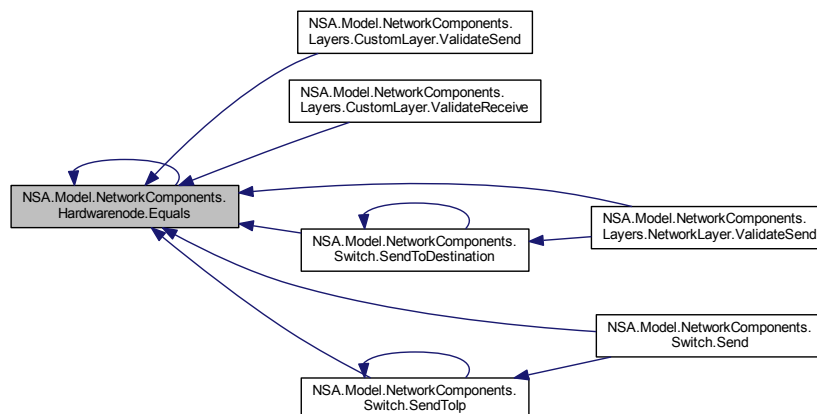
### Rückgabe

`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:





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

Equalses the specified other.

Parameter

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

Rückgabe

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

Gets the connection at port.

Parameter

<i>IfaceName</i>	Name of the port.
------------------	-------------------

Rückgabe

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

Returns a hash code for this instance.

Rückgabe

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

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

Gets the interface count.

Rückgabe

int: interface count

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

Gets the new interface number.

Rückgabe

int: number for next interface

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

Gets the port index of connection.

## Parameter

<i>C</i>	The connection.
----------	-----------------

## Rückgabe

Portindex

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



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

Determines if there is an [Interface](#) with the specified name.

## Parameter

<i>IfaceName</i>	Name of the iface.
------------------	--------------------

## Rückgabe

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

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

## Parameter

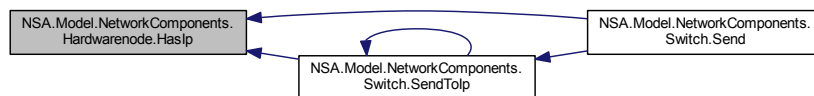
<i>Ip</i>	The ip.
-----------	---------

## Rückgabe

A bool

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

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



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

Interfaces the is used.

Parameter

<i>InterfaceName</i>	Name of the interface.
----------------------	------------------------

Rückgabe

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



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

Implements the operator !=.

Parameter

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

Rückgabe

The result of the operator.

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

Implements the operator ==.

Parameter

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

Rückgabe

The result of the operator.

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

[Hardwarenode](#) receives the package.

Parameter

<i>Tags</i>	Optional tags.
<i>ValInfo</i>	The validation Info
<i>Destination</i>	The destination.

Rückgabe

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

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

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

Removes a connection.

Parameter

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

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

Removes the interface with the given name.

Parameter

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

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

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

#### Parameter

<i>Destination</i>	The destination.
<i>Tags</i>	Optional tags.
<i>VallInfo</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](#).

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

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

## 6.5.4 Dokumentation der Property's

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

Gets or sets the connections.

The connections.

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

Gets or sets the interfaces.

The interfaces.

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

Gets the layerstack.

The layerstack.

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

Gets or sets the name of the hardwarenode.

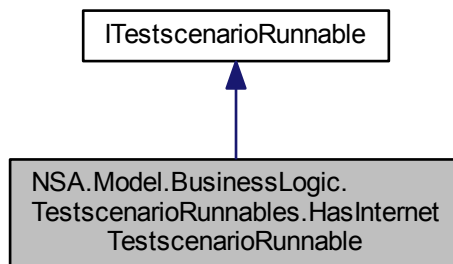
The name.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

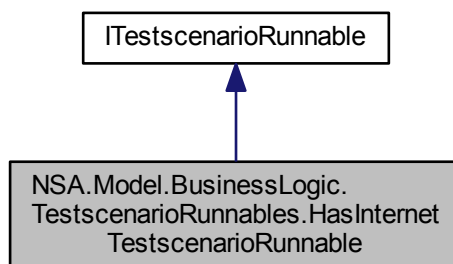
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/[Hardwarenode.cs](#)

## 6.6 NSA.Model.BusinessLogic.TestscenarioRunnables.HasInternetTestscenarioRunnable Klassenreferenz

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



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



## Öffentliche Methoden

- [HasInternetTestscenarioRunnable](#) ([Rule Rule](#))  
*Initializes a new instance of the [NSA.Model.BusinessLogic.TestscenarioRunnables.HasInternetTestscenarioRunnable](#) class.*
- `List< Simulation > Run ()`  
*runs all simulations for a given rule*

### 6.6.1 Beschreibung der Konstruktoren und Destruktoren

#### 6.6.1.1 NSA.Model.BusinessLogic.TestscenarioRunnables.HasInternetTestscenarioRunnable ( [Rule Rule](#) )

Initializes a new instance of the [NSA.Model.BusinessLogic.TestscenarioRunnables.HasInternetTestscenarioRunnable](#) class.

Parameter

<a href="#">Rule</a>	<a href="#">Rule</a> object
----------------------	-----------------------------

### 6.6.2 Dokumentation der Elementfunktionen

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

runs all simulations for a given rule

Rückgabe

simulations that failed

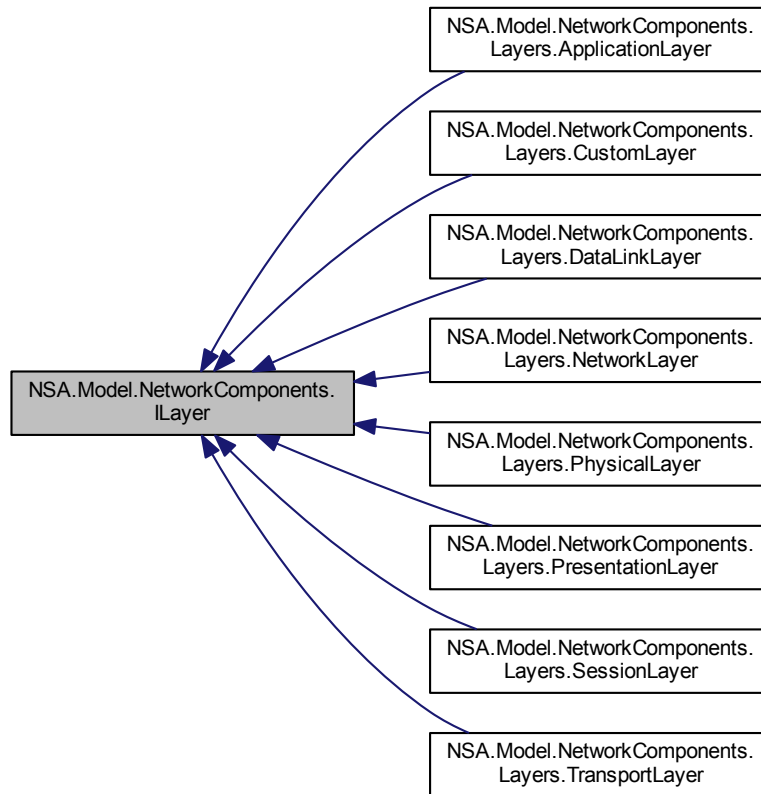
Implementiert [NSA.Model.BusinessLogic.TestscenarioRunnables.ITestscenarioRunnable](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

## 6.7 NSA.Model.NetworkComponents.ILayer Schnittstellenreferenz

Klassendiagramm für NSA.Model.NetworkComponents.ILayer:



### Öffentliche Methoden

- void `ValidateSend` (`Workstation` Destination, `Workstation` CurrentNode, `ValidationInfo` ValInfo, Dictionary< string, object > Tags, int LayerIndex)  
*Validates the layer while sending a packet.*
- bool `ValidateReceive` (`Workstation` CurrentNode, `ValidationInfo` ValInfo, Dictionary< string, object > Tags, `Hardwarenode` Destination, int LayerIndex)  
*Validates the layer while receiving a packet.*
- string `GetLayerName` ()  
*Gets the name of the layer.*
- bool `SetLayerName` (string NewName)  
*Sets the name of the layer.*
- int `GetLayerIndex` ()  
*Gets the index of the layer.*
- void `SetLayerIndex` (int I)  
*Sets the index of the layer.*



### 6.7.1 Dokumentation der Elementfunktionen

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

Gets the index of the layer.

Rückgabe

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

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



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

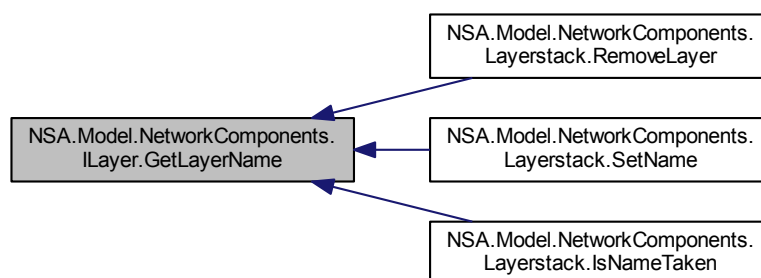
Gets the name of the layer.

Rückgabe

The Layername

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

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



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

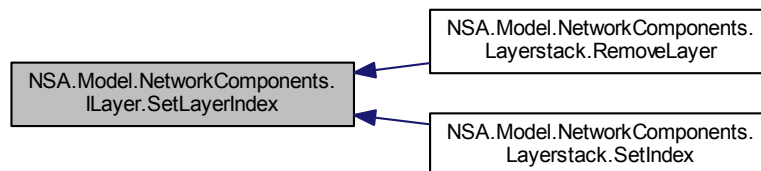
Sets the index of the layer.

#### Parameter

I	The Index.
---	------------

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

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



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

Sets the name of the layer.

#### Parameter

<i>NewName</i>	
----------------	--

#### Rückgabe

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

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



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

Validates the layer while receiving a packet.

Parameter

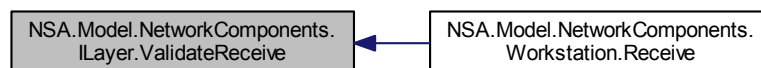
<i>CurrentNode</i>	
<i>ValInfo</i>	
<i>Tags</i>	
<i>Destination</i>	
<i>LayerIndex</i>	

Rückgabe

Boolean value indicating if the validation was successfull

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

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



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

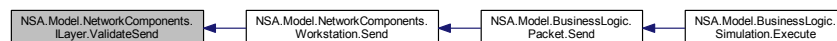
Validates the layer while sending a packet.

## Parameter

<i>Destination</i>	
<i>CurrentNode</i>	
<i>VallInfo</i>	
<i>Tags</i>	
<i>LayerIndex</i>	

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

## 6.8 NSA.Model.NetworkComponents.Interface Klassenreferenz

Class for a interface of an hardwarenode.

### Öffentliche Methoden

- [Interface](#) (IPAddress Ip, IPAddress Mask, int Number)  
*Initializes a new instance of the [Interface](#) class.*
- void [SetInterface](#) (IPAddress Ip, IPAddress Mask)  
*Sets the interface.*

### Öffentliche Attribute

- const string [NamePrefix](#) = "eth"  
*The name prefix of the interfacename.*

### Propertys

- string [Name](#) [get]  
*Gets the name of the interface*
- IPAddress [IpAddress](#) [get, set]  
*Gets or sets the ip address.*
- IPAddress [Subnetmask](#) [get, set]  
*Gets or sets the subnetmask.*

### 6.8.1 Ausführliche Beschreibung

Class for a interface of an hardwarenode.

### 6.8.2 Beschreibung der Konstruktoren und Destruktoren

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

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

Parameter

<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).

### 6.8.3 Dokumentation der Elementfunktionen

#### 6.8.3.1 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.

### 6.8.4 Dokumentation der Datenelemente

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

The name prefix of the interfacename.

### 6.8.5 Dokumentation der Propertys

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

Gets or sets the ip address.

The ip address.

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

Gets the name of the interface

The name.

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

Gets or sets the subnetmask.

The subnetmask.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

## 6.9 NSA.Model.NetworkComponents.Helper\_Classes.IPAddressExtensions Klassenreferenz

### Öffentliche, statische Methoden

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

#### 6.9.1 Dokumentation der Elementfunktionen

6.9.1.1 static IPAddress NSA.Model.NetworkComponents.Helper\_Classes.IPAddressExtensions.GetBroadcastAddress ( this IPAddress address, IPAddress subnetMask ) [static]

6.9.1.2 static IPAddress NSA.Model.NetworkComponents.Helper\_Classes.IPAddressExtensions.GetNetworkAddress ( this IPAddress address, IPAddress subnetMask ) [static]

6.9.1.3 static bool NSA.Model.NetworkComponents.Helper\_Classes.IPAddressExtensions.IsInSameSubnet ( this IPAddress address2, IPAddress address, IPAddress subnetMask ) [static]

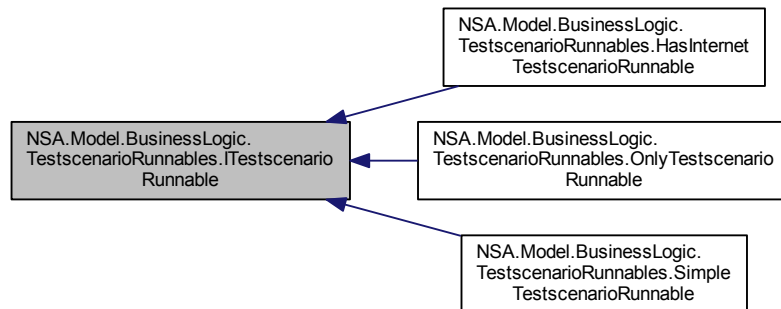
6.9.1.4 static bool NSA.Model.NetworkComponents.Helper\_Classes.IPAddressExtensions.IsValidSubnetMask ( this IPAddress Subnetmask ) [static]

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

## 6.10 NSA.Model.BusinessLogic.TestscenarioRunnables.ITestscenarioRunnable Schnittstellenreferenz

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



### Öffentliche Methoden

- `List< Simulation > Run ()`  
*runs all simulations for a given rule*

### 6.10.1 Dokumentation der Elementfunktionen

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

runs all simulations for a given rule

#### Rückgabe

simulations that failed

Implementiert in [NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable](#), [NSA.Model.BusinessLogic.TestscenarioRunnables.HasInternetTestscenarioRunnable](#) und [NSA.Model.BusinessLogic.TestscenarioRunnables.SimpleTestscenarioRunnable](#).

Die Dokumentation für diese Schnittstelle wurde erzeugt aufgrund der Datei:

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

## 6.11 NSA.Model.NetworkComponents.Layerstack Klassenreferenz

### Layerstack

## Öffentliche Methoden

- [Layerstack](#) ()  
*Initializes a new instance of the [Layerstack](#) class.*
- void [AddLayer](#) (ILayer Lay)  
*Adds a layer to the stack.*
- void [RemoveLayer](#) (string Name)  
*Removes a layer from the stack.*
- int [GetSize](#) ()  
*Returns the size of the layerstack.*
- ILayer [GetLayer](#) (int Index)  
*Returns the layer at the index.*
- void [InsertAt](#) (int Index, ILayer Layer)  
*Inserts a layer at the index.*
- void [SetIndex](#) (string Name, int NewIndex)  
*Sets the index.*
- ILayer [GetLayerByName](#) (string Name)  
*Gets the layer with the name.*
- bool [SetName](#) (string OldName, string NewName)  
*Sets the name.*
- List< ILayer > [GetAllLayers](#) ()  
*Gets all layers.*
- bool [IsNameTaken](#) (string Name)  
*Determines whether the name is taken or not.*
- string [CreateUniqueName](#) ()  
*Creates a new unique name.*

### 6.11.1 Ausführliche Beschreibung

#### [Layerstack](#)

### 6.11.2 Beschreibung der Konstruktoren und Destruktoren

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

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

### 6.11.3 Dokumentation der Elementfunktionen

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

Adds a layer to the stack.

##### Parameter

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



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



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

Creates a new unique name.

**Rückgabe**

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

Gets all layers.

**Rückgabe**

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

Returns the layer at the index.

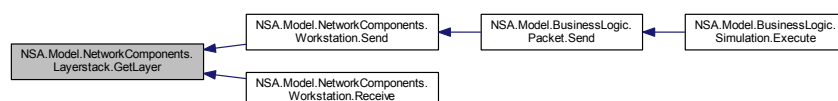
**Parameter**

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

**Rückgabe**

The layer

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



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

Gets the layer with the name.

##### Parameter

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

##### Rückgabe

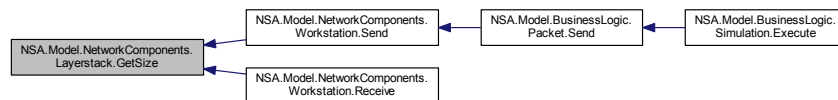
#### 6.11.3.6 `int NSA.Model.NetworkComponents.Layerstack.GetSize ( )`

Returns the size of the layerstack.

##### Rückgabe

The size

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



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

Inserts a layer at the index.

##### Parameter

<i>Index</i>	The index.
<i>Layer</i>	The layer.

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

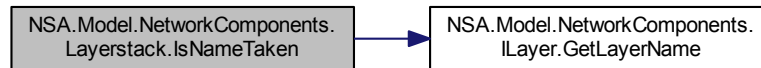
Determines whether the name is taken or not.

##### Parameter

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

Rückgabe

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



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

Removes a layer from the stack.

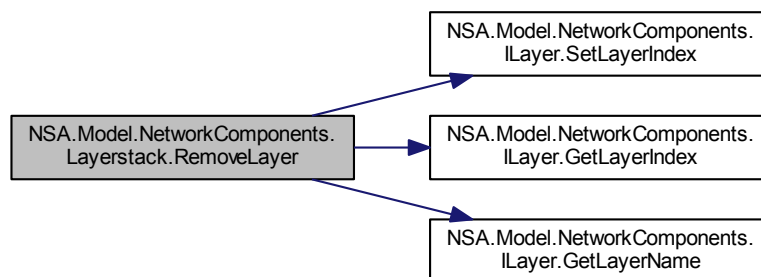
Parameter

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

Ausnahmebehandlung

<i>System.InvalidOperationException</i>	Layer with the name: + name + does not exist.
-----------------------------------------	-----------------------------------------------

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



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

Sets the index.

## Parameter

<i>Name</i>	The name.
<i>NewIndex</i>	The new index.

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



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

Sets the name.

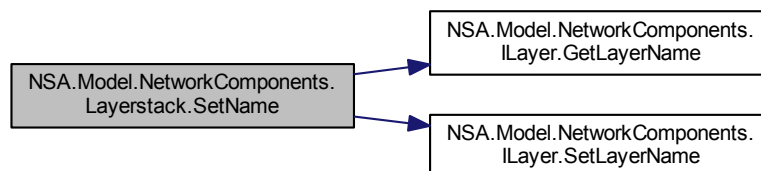
## Parameter

<i>OldName</i>	The old name.
<i>NewName</i>	The new name.

## Rückgabe

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

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



Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

## 6.12 NSA.Model.NetworkComponents.Network Klassenreferenz

### Öffentliche Methoden

- [Network](#) ()
- [Hardwarenode GetHardwarenodeByName](#) (string Name)  
*Returns the [Hardwarenode](#) with the name.*
- void [AddHardwarenode](#) ([Hardwarenode](#) NewNode)  
*Adds a hardwarenode.*
- bool [AddConnection](#) (string StartNodeInterfaceName, string EndNodeInterfaceName, [Connection](#) New↔ Connection)  
*Adds the connection.*
- void [RemoveHardwarnode](#) (string Name)  
*Removes the hardwarnode.*
- void [RemoveConnection](#) (string ConnectionName)  
*Removes the connection.*
- [Hardwarenode GetWorkstationByIp](#) (IPAddress Ip)  
*Gets the workstation by ip.*
- List< [Hardwarenode](#) > [GetAllHardwarenodes](#) ()  
*Gets all hardwarenodes.*
- List< [Workstation](#) > [GetAllWorkstations](#) ()  
*Gets all workstations.*
- [Connection GetConnectionByName](#) (string Name)  
*Gets the name of the connection by.*
- List< [Router](#) > [GetRouters](#) ()  
*Gets the routers with internetconnection.*
- List< [Connection](#) > [GetAllConnections](#) ()
- List< [Workstation](#) > [GetHardwareNodesForSubnet](#) (string Subnetmask)  
*Gets all hardwarenodes belonging to a subnet.*

### Propertys

- List< [Connection](#) > [Connections](#) [get]

### 6.12.1 Beschreibung der Konstruktoren und Destruktoren

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

### 6.12.2 Dokumentation der Elementfunktionen

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

Adds the connection.

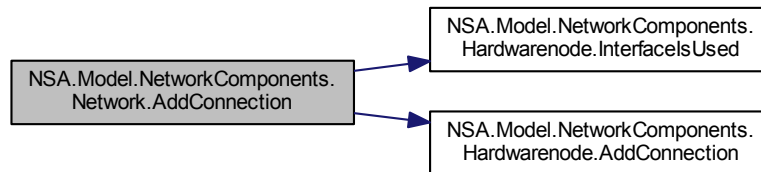
#### Parameter

<i>StartNodeInterfaceName</i>	Start name of the node interface.
<i>EndNodeInterfaceName</i>	End name of the node interface.
<i>NewConnection</i>	The new connection.

**Rückgabe**

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

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



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

Adds a hardwarenode.

**Parameter**

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

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

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

Gets all hardwarenodes.

**Rückgabe**

all Hardwarenodes

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

Gets all workstations.

**Rückgabe**

all Workstations

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

Gets the name of the connection by.

## Parameter

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

## Rückgabe

the connection with its name

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

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

## Parameter

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

## Rückgabe

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

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

Gets all hardwarenodes belonging to a subnet.

## Parameter

<i>Subnetmask</i>	The subnetmask.
-------------------	-----------------

## Rückgabe

A list of hardwarenodes who belong to the subnet.

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

Gets the routers with internetconnection.

## Rückgabe

A List of routers

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

Gets the workstation by ip.

## Parameter

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

## Rückgabe

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

Removes the connection.

## Parameter

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

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

Removes the hardwarnode.

## Parameter

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

## 6.12.3 Dokumentation der Propertys

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

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

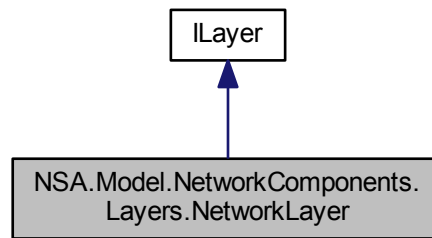
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/[Network.cs](#)

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

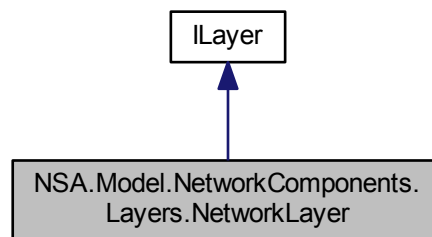
Network-Layer



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



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



## Öffentliche Methoden

- `NetworkLayer` (int I)  
*Initializes a new instance of the `NetworkLayer` class.*
- bool `ValidateReceive` (`Workstation` CurrentNode, `ValidationInfo` ValInfo, Dictionary< string, object > Tags, `Hardwarenode` Destination, int LayerIndex)  
*Validates the layer while receiving a packet.*
- string `GetLayerName` ()  
*Gets the name of the layer.*
- bool `SetLayerName` (string NewName)  
*Sets the name of the layer.*
- int `GetLayerIndex` ()  
*Gets the index of the layer.*
- void `SetLayerIndex` (int I)  
*Sets the index of the layer.*
- void `ValidateSend` (`Workstation` Destination, `Workstation` CurrentNode, `ValidationInfo` ValInfo, Dictionary< string, object > Tags, int LayerIndex)  
*Validates the layer while sending a packet.*

### 6.13.1 Ausführliche Beschreibung

Network-Layer

Siehe auch

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

### 6.13.2 Beschreibung der Konstruktoren und Destruktoren

#### 6.13.2.1 NSA.Model.NetworkComponents.Layers.NetworkLayer.NetworkLayer ( int / )

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

Parameter

/	The index.
---	------------

### 6.13.3 Dokumentation der Elementfunktionen

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

Gets the index of the layer.

Rückgabe

The index

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

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

Gets the name of the layer.

Rückgabe

The Layername

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

#### 6.13.3.3 void NSA.Model.NetworkComponents.Layers.NetworkLayer.SetLayerIndex ( int / )

Sets the index of the layer.

## Parameter

/	The Index.
---	------------

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

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

Sets the name of the layer.

## Parameter

<i>NewName</i>	New Name
----------------	----------

## Rückgabe

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

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

Validates the layer while receiving a packet.

## Parameter

<i>CurrentNode</i>	Current node
<i>VallInfo</i>	Validation Info
<i>Tags</i>	Tags
<i>Destination</i>	Destinationnode
<i>LayerIndex</i>	Index of the Layer

## Rückgabe

Boolean value indicating if the validation was successfull

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

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

Validates the layer while sending a packet.

## Parameter

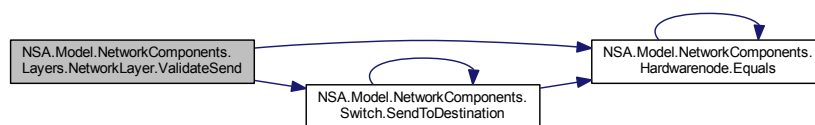
<i>Destination</i>	The Destination
--------------------	-----------------

## Parameter

<i>CurrentNode</i>	Current Node
<i>VallInfo</i>	Validation Info
<i>Tags</i>	Tags
<i>LayerIndex</i>	The Layer index

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

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

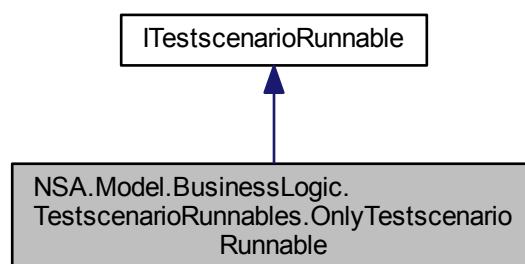


Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

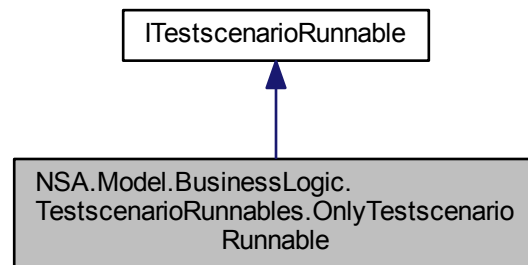
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Layers/[NetworkLayer.cs](#)

## 6.14 NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable Klassenreferenz

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



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



## Öffentliche Methoden

- [OnlyTestscenarioRunnable](#) ([Rule Rule](#), [Network N](#))  
*Initializes a new instance of the [NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable](#) class.*
- List< [Simulation](#) > [Run](#) ()  
*runs all simulations for a given rule*

### 6.14.1 Beschreibung der Konstruktoren und Destruktoren

#### 6.14.1.1 NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable.OnlyTestscenarioRunnable ( [Rule Rule](#), [Network N](#) )

Initializes a new instance of the [NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable](#) class.

Parameter

<a href="#">Rule</a>	<a href="#">Rule</a> object
<a href="#">N</a>	Network

### 6.14.2 Dokumentation der Elementfunktionen

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

runs all simulations for a given rule

**Rückgabe**

simulations that failed

Implementiert [NSA.Model.BusinessLogic.TestscenarioRunnables.ITestscenarioRunnable](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

## 6.15 NSA.Model.BusinessLogic.Packet Klassenreferenz

Class for a packet

**Öffentliche Methoden**

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

**Propertyys**

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

### 6.15.1 Ausführliche Beschreibung

Class for a packet

### 6.15.2 Beschreibung der Konstruktoren und Destruktoren

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

Constructor for [Packet](#)

## Parameter

<i>Src</i>	Source-Node
<i>Dest</i>	Destination-Node
<i>T</i>	TTL
<i>ExpRes</i>	Expected Result

## 6.15.3 Dokumentation der Elementfunktionen

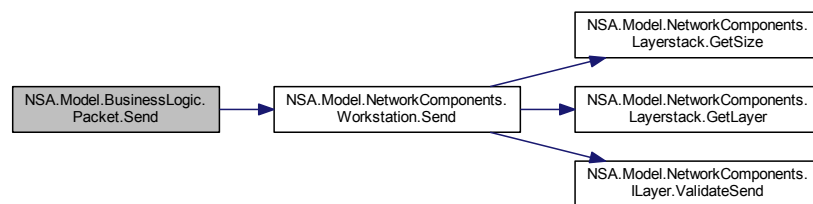
## 6.15.3.1 Packet NSA.Model.BusinessLogic.Packet.Send ( )

Sends this packet to the destination.

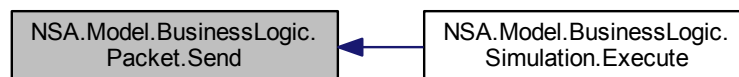
## Rückgabe

The Returnpacket if sending to destination was successfull or null

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



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



## 6.15.4 Dokumentation der Propertys

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

Returns the Destination-Node

The destination.

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

Returns the expected result.

Expected Result

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

Returns the Hops of the packet

The Hops

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

Returns the result of the packet

Result of the packet

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

Returns the Source-Node

The source

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

The Time-To-Life

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

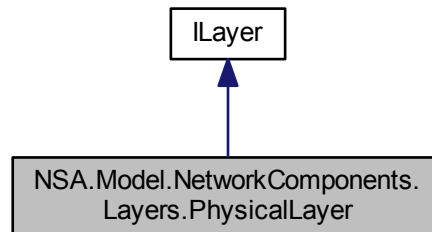
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/BusinessLogic/[Packet.cs](#)



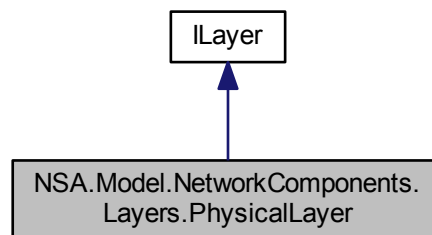
## 6.16 NSA.Model.NetworkComponents.Layers.PhysicalLayer Klassenreferenz

Physical-Layer

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



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



### Öffentliche Methoden

- `PhysicalLayer` (int I)  
*Initializes a new instance of the `PhysicalLayer` class.*
- bool `ValidateReceive` (Workstation CurrentNode, ValidationInfo ValInfo, Dictionary< string, object > Tags, Hardwarenode Destination, int LayerIndex)  
*Validates the layer while receiving a packet.*
- string `GetLayerName` ()  
*Gets the name of the layer.*
- bool `SetLayerName` (string NewName)  
*Sets the name of the layer.*
- int `GetLayerIndex` ()  
*Gets the index of the layer.*
- void `SetLayerIndex` (int I)

*Sets the index of the layer.*

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

*Validates the layer while sending a packet.*

### 6.16.1 Ausführliche Beschreibung

Physical-Layer

Siehe auch

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

### 6.16.2 Beschreibung der Konstruktoren und Destruktoren

#### 6.16.2.1 [NSA.Model.NetworkComponents.Layers.PhysicalLayer.PhysicalLayer](#) ( int / )

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

Parameter

/	The index.
---	------------

### 6.16.3 Dokumentation der Elementfunktionen

#### 6.16.3.1 int [NSA.Model.NetworkComponents.Layers.PhysicalLayer.GetLayerIndex](#) ( )

Gets the index of the layer.

Rückgabe

The index

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

#### 6.16.3.2 string [NSA.Model.NetworkComponents.Layers.PhysicalLayer.GetLayerName](#) ( )

Gets the name of the layer.

Rückgabe

The Layername

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

#### 6.16.3.3 void [NSA.Model.NetworkComponents.Layers.PhysicalLayer.SetLayerIndex](#) ( int / )

Sets the index of the layer.

## Parameter

/	The Index.
---	------------

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

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

Sets the name of the layer.

## Parameter

<i>NewName</i>	New Name
----------------	----------

## Rückgabe

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

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

Validates the layer while receiving a packet.

## Parameter

<i>CurrentNode</i>	Current node
<i>VallInfo</i>	Validation Info
<i>Tags</i>	Tags
<i>Destination</i>	Destinationnode
<i>LayerIndex</i>	Index of the Layer

## Rückgabe

Boolean value indicating if the validation was successfull

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

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

Validates the layer while sending a packet.

## Parameter

<i>Destination</i>	The Destination
--------------------	-----------------

## Parameter

<i>CurrentNode</i>	Current Node
<i>ValInfo</i>	Validation Info
<i>Tags</i>	Tags
<i>LayerIndex</i>	The Layer index

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

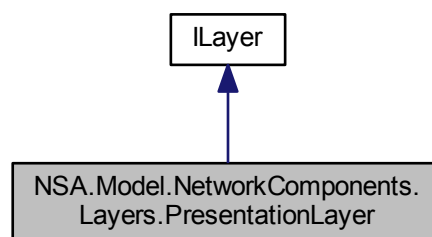
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

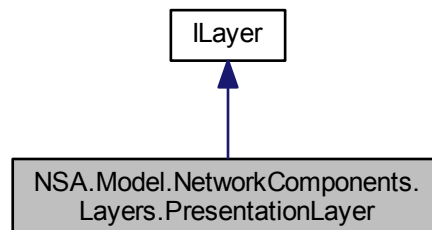
## 6.17 NSA.Model.NetworkComponents.Layers.PresentationLayer Klassenreferenz

Presentation-Layer

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



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



## Öffentliche Methoden

- [PresentationLayer](#) (int I)  
*Initializes a new instance of the [PresentationLayer](#) class.*
- bool [ValidateReceive](#) ([Workstation](#) CurrentNode, [ValidationInfo](#) ValInfo, Dictionary< string, object > Tags, [Hardwarenode](#) Destination, int LayerIndex)  
*Validates the layer while receiving a packet.*
- string [GetLayerName](#) ()  
*Gets the name of the layer.*
- bool [SetLayerName](#) (string NewName)  
*Sets the name of the layer.*
- int [GetLayerIndex](#) ()  
*Gets the index of the layer.*
- void [SetLayerIndex](#) (int I)  
*Sets the index of the layer.*
- void [ValidateSend](#) ([Workstation](#) Destination, [Workstation](#) CurrentNode, [ValidationInfo](#) ValInfo, Dictionary< string, object > Tags, int LayerIndex)  
*Validates the layer while sending a packet.*

### 6.17.1 Ausführliche Beschreibung

Presentation-Layer

Siehe auch

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

### 6.17.2 Beschreibung der Konstruktoren und Destruktoren

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

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

Parameter

I	The index.
---	------------

### 6.17.3 Dokumentation der Elementfunktionen

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

Gets the index of the layer.

Rückgabe

Index

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

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

Gets the name of the layer.

#### Rückgabe

The Layername

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

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

Sets the index of the layer.

#### Parameter

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

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

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

Sets the name of the layer.

#### Parameter

<i>NewName</i>	New Name
----------------	----------

#### Rückgabe

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

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

Validates the layer while receiving a packet.

#### Parameter

<i>CurrentNode</i>	Current node
<i>VallInfo</i>	Validation Info
<i>Tags</i>	Tags
<i>Destination</i>	Destinationnode
<i>LayerIndex</i>	Index of the Layer

**Rückgabe**

Boolean value indicating if the validation was successfull

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

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

Validates the layer while sending a packet.

**Parameter**

<i>Destination</i>	The Destination
<i>CurrentNode</i>	Current Node
<i>VallInfo</i>	Validation Info
<i>Tags</i>	Tags
<i>LayerIndex</i>	The Layer index

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

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

## 6.18 NSA.Model.BusinessLogic.Project Klassenreferenz

Class for project

**Öffentliche Methoden**

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

**Propertys**

- string [Path](#) [get, set]  
*Gets or sets the path.*
- [Network Network](#) [get, set]  
*Gets or sets the network.*

### 6.18.1 Ausführliche Beschreibung

Class for project

## 6.18.2 Beschreibung der Konstruktoren und Destruktoren

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

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

## 6.18.3 Dokumentation der Propertys

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

Gets or sets the network.

The network.

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

Gets or sets the path.

The path.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

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

[Result](#) class for the packetresult

### Öffentliche Typen

- enum [Errors](#) {  
[Errors.NoError](#), [Errors.NoRoute](#), [Errors.NoConnection](#), [Errors.PacketNotForThisNode](#),  
[Errors.SwitchNoConnection](#), [Errors.SourceDestinationNull](#), [Errors.TtlError](#), [Errors.CustomLayerError](#),  
[Errors.CustomLayerIndexError](#), [Errors.NoPackets](#) }

*Enum for the possible errors*

### Öffentliche Methoden

- [Result](#) ()  
*Default constructor*
- [Result](#) ([Errors](#) ErrId, string R, [ILayer](#) L)  
*Constructor*



## Statische öffentliche Attribute

- static readonly string[] [ResultStrings](#)  
*The possible result strings*

## Propertys

- [Errors ErrorId](#) [get, set]  
*Gets or sets the error identifier.*
- string [Res](#) [get, set]  
*Gets or sets the result string.*
- [ILayer LayerError](#) [get, set]  
*Gets or sets the layer of the error.*
- bool [SendError](#) [get, set]  
*Gets or sets a value indicating whether it's a send error.*

### 6.19.1 Ausführliche Beschreibung

[Result](#) class for the packetresult

### 6.19.2 Dokumentation der Aufzählungstypen

6.19.2.1 enum [NSA.Model.NetworkComponents.Helper\\_Classes.Result.Errors](#) [strong]

Enum for the possible errors

#### Aufzählungswerte

- NoError** No error
- NoRoute** No [Route](#) or Standard-Gateway was found
- NoConnection** No connection to the next node
- PacketNotForThisNode** The packet was not for this node
- SwitchNoConnection** The switch has no connection to the next node
- SourceDestinationNull** The source or destination is null
- TtlError** The TTL is zero
- CustomLayerError** Custom Layer Error
- CustomLayerIndexError** Custom layer index error
- NoPackets** No Packets in the simulation

### 6.19.3 Beschreibung der Konstruktoren und Destruktoren

#### 6.19.3.1 NSA.Model.NetworkComponents.Helper\_Classes.Result.Result ( )

Default constructor

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



#### 6.19.3.2 NSA.Model.NetworkComponents.Helper\_Classes.Result.Result ( Errors *ErrId*, string *R*, ILayer *L* )

Constructor

Parameter

<i>ErrId</i>	The error identifier.
<i>R</i>	The result string.
<i>L</i>	The layer.

### 6.19.4 Dokumentation der Datenelemente

#### 6.19.4.1 readonly string [ ] NSA.Model.NetworkComponents.Helper\_Classes.Result.ResultStrings [static]

Initialisierung:

```

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

The possible result strings

## 6.19.5 Dokumentation der Propertys

### 6.19.5.1 Errors NSA.Model.NetworkComponents.Helper\_Classes.Result.ErrorId [get], [set]

Gets or sets the error identifier.

The error identifier.

### 6.19.5.2 ILayer NSA.Model.NetworkComponents.Helper\_Classes.Result.LayerError [get], [set]

Gets or sets the layer of the error.

The layer of the error.

### 6.19.5.3 string NSA.Model.NetworkComponents.Helper\_Classes.Result.Res [get], [set]

Gets or sets the result string.

The result string

### 6.19.5.4 bool NSA.Model.NetworkComponents.Helper\_Classes.Result.SendError [get], [set]

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

true if send error otherwise, false.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

## 6.20 NSA.Model.NetworkComponents.Route Klassenreferenz

class for a single route of the routingtable

### Öffentliche Methoden

- [Route](#) (IPAddress [Destination](#), IPAddress [Subnetmask](#), IPAddress [Gateway](#), [Interface Iface](#))  
*Initializes a new instance of the [Route](#) class.*
- void [SetRoute](#) (IPAddress DestinationIp, IPAddress Mask, IPAddress GatewayAddress, [Interface](#) Intface)  
*Sets the route.*

## Propertys

- string [Name](#) [get]  
*Gets the name (id) of the route. Every route has a unique id.*
- IPAddress [Destination](#) [get]  
*Gets the destination IP.*
- IPAddress [Subnetmask](#) [get]  
*Gets the subnetmask.*
- IPAddress [Gateway](#) [get]  
*Gets the gateway.*
- [Interface Iface](#) [get]  
*Gets the interface.*

### 6.20.1 Ausführliche Beschreibung

class for a single route of the routingtable

### 6.20.2 Beschreibung der Konstruktoren und Destruktoren

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

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

Parameter

<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.

### 6.20.3 Dokumentation der Elementfunktionen

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

Sets the route.

Parameter

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

## 6.20.4 Dokumentation der Propertys

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

Gets the destination IP.

The destination.

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

Gets the gateway.

The gateway.

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

Gets the interface.

The interface.

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

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

The name.

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

Gets the subnetmask.

The subnetmask.

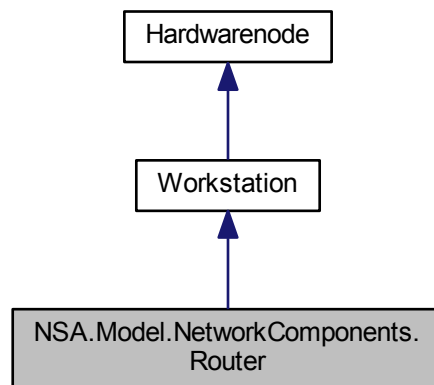
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

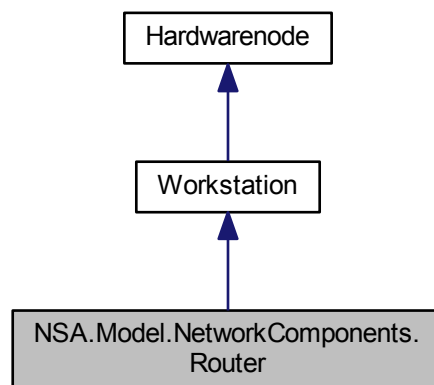
## 6.21 NSA.Model.NetworkComponents.Router Klassenreferenz

Implements the network component [Router](#).

Klassendiagramm für NSA.Model.NetworkComponents.Router:



Zusammengehörigkeiten von NSA.Model.NetworkComponents.Router:



### Öffentliche Methoden

- [Router](#) (string [Name](#))

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

## Propertys

- bool [IsGateway](#) [get, set]

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

## Weitere Geerbte Elemente

### 6.21.1 Ausführliche Beschreibung

Implements the network component [Router](#).

Siehe auch

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

### 6.21.2 Beschreibung der Konstruktoren und Destruktoren

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

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

Parameter

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

### 6.21.3 Dokumentation der Propertys

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

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

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

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

## 6.22 NSA.Model.BusinessLogic.Rule Klassenreferenz

[Rule](#) class parses the text and if it is valid, the [Rule](#) object will be used for simulations

## Öffentliche Methoden

- [Rule](#) (string [StartNode](#), List< string > [EndNodes](#), Dictionary< string, int > [Options](#), [SimulationType](#) [SimulationType](#), bool [ExpectedResult](#), [Network](#) N)  
*Initializes a new instance of the [NSA.Model.BusinessLogic.Rule](#) class.*

## Öffentliche, statische Methoden

- static [Rule](#) [Parse](#) (string [Rule](#), [Network](#) N)  
*parses a string and creates a rule from it*
- static bool [CheckForTrueOrFalse](#) (string [Text](#), string [Rule](#))  
*Checks if text contains TRUE or FALSE*

## Statische öffentliche Attribute

- static List< string > [Parameters](#)  
*Accepted parameter types*

## Property

- [SimulationType](#) [SimulType](#) [get]  
*Gets the type of the simul.*
- string [StartNodeString](#) [get]  
*String representation of startNode*
- [Hardwarenode](#) [StartNode](#) [get]  
*Gets the start node.*
- List< string > [EndNodesString](#) [get]  
*String representation of endNodes*
- List< [Hardwarenode](#) > [EndNodes](#) [get]  
*Gets the end nodes.*
- Dictionary< string, int > [Options](#) [get]  
*[Rule](#) options*
- bool [ExpectedResult](#) [get]  
*Expected result for the [Rule](#)*

### 6.22.1 Ausführliche Beschreibung

[Rule](#) class parses the text and if it is valid, the [Rule](#) object will be used for simulations

Expected input: PC\_NAME | (PC\_NAME, ...) | { OPTIONS } | TRUE/FALSE PC\_NAME | (SUBNET(PC\_NAME), ...) | {TTL: 64, SSL: TRUE, ...} | TRUE/FALSE PC\_NAME | ONLY(PC\_NAME, ...) | {TTL: 64, SSL: TRUE, ...} | TRUE/FALSE PC\_NAME | HAS\_INTERNET | TRUE/FALSE

### 6.22.2 Beschreibung der Konstruktoren und Destruktoren

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

Initializes a new instance of the [NSA.Model.BusinessLogic.Rule](#) class.



## Parameter

<i>StartNode</i>	Start node.
<i>EndNodes</i>	End nodes.
<i>Options</i>	Options.
<i>SimulationType</i>	<a href="#">Simulation</a> type.
<i>ExpectedResult</i>	the expected result: True or False
<i>N</i>	Network

## 6.22.3 Dokumentation der Elementfunktionen

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

Checks if text contains TRUE or FALSE

## Rückgabe

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

## Parameter

<i>Text</i>	text to parse
<i>Rule</i>	<a href="#">Rule</a> string, which is needed, in order to inform the user about invalid <a href="#">Rule</a> input

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

parses a string and creates a rule from it

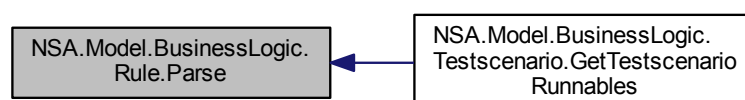
## Parameter

<i>Rule</i>	
<i>N</i>	

## Rückgabe

[Rule](#) object

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



## 6.22.4 Dokumentation der Datenelemente

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

#### Initialisierung:

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

Accepted parameter types

## 6.22.5 Dokumentation der Propertys

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

Gets the end nodes.

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

String representation of endNodes

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

Expected result for the [Rule](#)

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

[Rule](#) options

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

Gets the type of the simul.

The type of the simul.

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

Gets the start node.

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

String representation of startNode

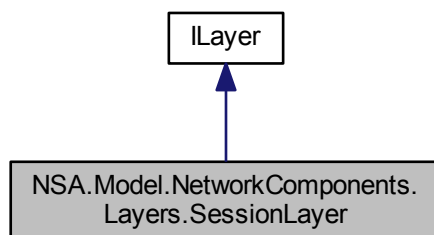
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

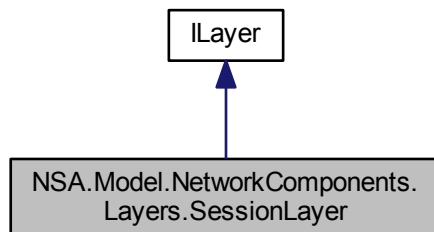
## 6.23 NSA.Model.NetworkComponents.Layers.SessionLayer Klassenreferenz

Session-Layer

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



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



## Öffentliche Methoden

- [SessionLayer](#) (int I)  
*Initializes a new instance of the [SessionLayer](#) class.*
- bool [ValidateReceive](#) ([Workstation](#) CurrentNode, [ValidationInfo](#) ValInfo, Dictionary< string, object > Tags, [Hardwarenode](#) Destination, int LayerIndex)  
*Validates the layer while receiving a packet.*
- string [GetLayerName](#) ()  
*Gets the name of the layer.*
- bool [SetLayerName](#) (string NewName)  
*Sets the name of the layer.*
- int [GetLayerIndex](#) ()  
*Gets the index of the layer.*
- void [SetLayerIndex](#) (int I)  
*Sets the index of the layer.*
- void [ValidateSend](#) ([Workstation](#) Destination, [Workstation](#) CurrentNode, [ValidationInfo](#) ValInfo, Dictionary< string, object > Tags, int LayerIndex)  
*Validates the layer while sending a packet.*

### 6.23.1 Ausführliche Beschreibung

Session-Layer

Siehe auch

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

### 6.23.2 Beschreibung der Konstruktoren und Destruktoren

#### 6.23.2.1 NSA.Model.NetworkComponents.Layers.SessionLayer.SessionLayer ( int I )

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

Parameter

I	The index.
---	------------

### 6.23.3 Dokumentation der Elementfunktionen

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

Gets the index of the layer.

Rückgabe

Index

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

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

Gets the name of the layer.

Rückgabe

The Layername

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

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

Sets the index of the layer.

Parameter

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

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

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

Sets the name of the layer.

Parameter

<i>NewName</i>	New Name
----------------	----------

Rückgabe

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

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

Validates the layer while receiving a packet.

Parameter

<i>CurrentNode</i>	Current node
<i>VallInfo</i>	Validation Info
<i>Tags</i>	Tags
<i>Destination</i>	Destinationnode
<i>LayerIndex</i>	Index of the Layer

**Rückgabe**

Boolean value indicating if the validation was successfull

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

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

Validates the layer while sending a packet.

**Parameter**

<i>Destination</i>	The Destination
<i>CurrentNode</i>	Current Node
<i>ValInfo</i>	Validation Info
<i>Tags</i>	Tags
<i>LayerIndex</i>	The Layer index

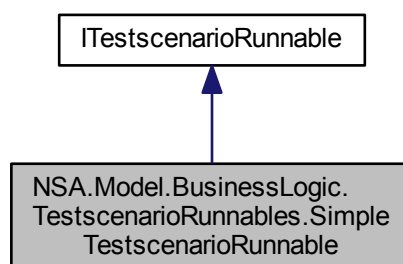
Implementiert [NSA.Model.NetworkComponents.ILayer](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

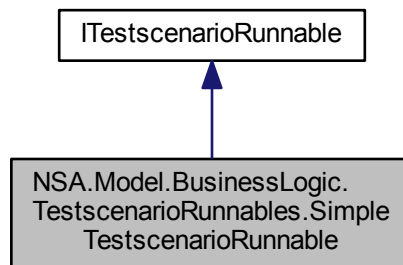
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/NSA.Model/NetworkComponents/Layers/[SessionLayer.cs](#)

## 6.24 NSA.Model.BusinessLogic.TestscenarioRunnables.SimpleTestscenarioRunnable Klassenreferenz

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



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



## Öffentliche Methoden

- [SimpleTestscenarioRunnable \(Rule Rule\)](#)  
*Initializes a new instance of the [NSA.Model.BusinessLogic.TestscenarioRunnables.SimpleTestscenarioRunnable](#) class.*
- List< [Simulation](#) > [Run](#) ()  
*runs all simulations for a given rule*

### 6.24.1 Beschreibung der Konstruktoren und Destruktoren

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

Initializes a new instance of the [NSA.Model.BusinessLogic.TestscenarioRunnables.SimpleTestscenarioRunnable](#) class.

Parameter

<a href="#">Rule</a>	<a href="#">Rule</a> object
----------------------	-----------------------------

### 6.24.2 Dokumentation der Elementfunktionen

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

runs all simulations for a given rule

**Rückgabe**

simulations that failed

Implementiert [NSA.Model.BusinessLogic.TestscenarioRunnables.ITestscenarioRunnable](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

**6.25 NSA.Model.BusinessLogic.Simulation Klassenreferenz**

Class for the simulation

**Öffentliche Methoden**

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

**Propertys**

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

**6.25.1 Ausführliche Beschreibung**

Class for the simulation

**6.25.2 Beschreibung der Konstruktoren und Destruktoren****6.25.2.1 NSA.Model.BusinessLogic.Simulation.Simulation ( string I )**

Constructor for the simulation used in testscenarios



## Parameter

<i>I</i>	The ID
----------	--------

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

Constructor used for normal simulation

## Parameter

<i>I</i>	ID
<i>S</i>	Name of sourcenode
<i>D</i>	Name of destinationnode
<i>ExpRes</i>	Expected result

## 6.25.3 Dokumentation der Elementfunktionen

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

Adds a packet to the sendpackets

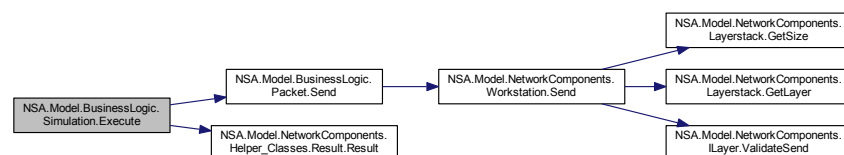
## Parameter

<i>Packet</i>	The packet to be added
---------------	------------------------

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

Executes this simulation

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



## 6.25.3.3 IEnumerable&lt;Packet&gt; NSA.Model.BusinessLogic.Simulation.GetAllPackets ( )

Returns all packets

## Rückgabe

All Packets

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

Returns the last packet

Rückgabe

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

### 6.25.4 Dokumentation der Propertys

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

Returns the name of the destinationnode

Name of the destinationnode

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

Returns the expected result

The expected result

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

Returns the ID

ID of the simulation

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

Returns the receivedpackets

The receivedpackets

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

Returns the sendpackets

The sendpackets

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

Returns the name of the sourcenode

Name of the sourcenode

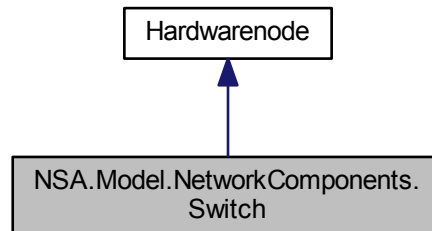
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

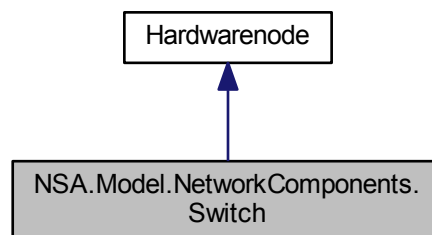
## 6.26 NSA.Model.NetworkComponents.Switch Klassenreferenz

Implements the network component switch.

Klassendiagramm für NSA.Model.NetworkComponents.Switch:



Zusammengehörigkeiten von NSA.Model.NetworkComponents.Switch:



### Öffentliche Methoden

- `Switch` (string `Name`)  
*Initializes a new instance of the `Switch` class.*
- override `Interface AddInterface` (IPAddress `Ip`, IPAddress `Subnetmask`, int `PortNum=-1`)  
*Adds a new interface with the given IP and subnetmask*
- List< `Interface` > `SetInterfaceCount` (int `Count`)  
*Sets the interface count to the given value.*
- override List< `Hardwarenode` > `Send` (`Hardwarenode` `Destination`, Dictionary< string, object > `Tags`, `ValidationInfo` `VallInfo`)  
*`Hardwarenode` sends the package to specified destination.*
- bool `SendToIp` (`ValidationInfo` `VallInfo`, `Connection` `ComingConn`)  
*Sends to ip.*
- bool `SendToDestination` (`Workstation` `Destination`, `ValidationInfo` `VallInfo`, `Connection` `ComingCon`, IPAddress `nodeIP`, IPAddress `subnetmask`)  
*Sends to destination.*

## Weitere Geerbte Elemente

### 6.26.1 Ausführliche Beschreibung

Implements the network component switch.

Siehe auch

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

### 6.26.2 Beschreibung der Konstruktoren und Destruktoren

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

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

Parameter

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

### 6.26.3 Dokumentation der Elementfunktionen

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

Adds a new interface with the given IP and subnetmask

Parameter

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

Rückgabe

The newly added [Interface](#)

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

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

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

## Parameter

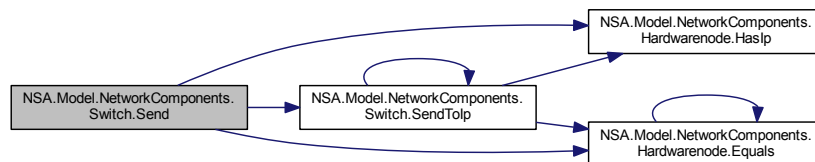
<i>Destination</i>	The destination.
<i>Tags</i>	Optional tags.
<i>ValInfo</i>	Validation Info

## 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:



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

Sends to destination.

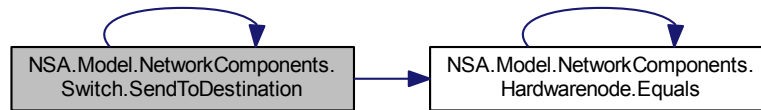
## Parameter

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

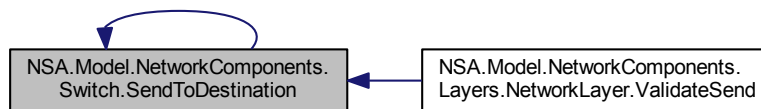
**Rückgabe**

Bool if it could send

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



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

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

Sends to ip.

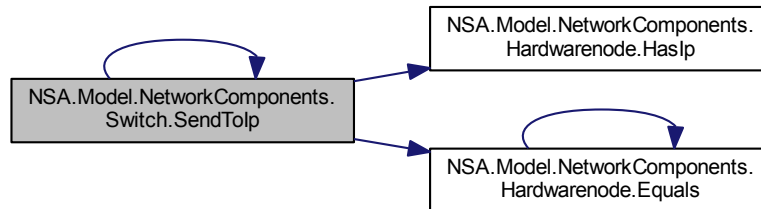
**Parameter**

<i>ValInfo</i>	The value information.
<i>ComingConn</i>	The coming connection.

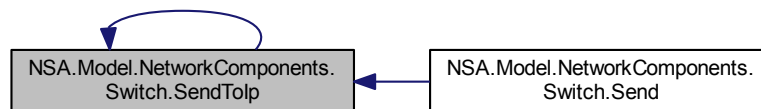
**Rückgabe**

Bool if it could send

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



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



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

Sets the interface count to the given value.

**Parameter**

<i>Count</i>	The count.
--------------	------------

**Rückgabe**

Returns a list of removed interfaces

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

## 6.27 NSA.Model.BusinessLogic.Testscenario Klassenreferenz

### Öffentliche Methoden

- [Testscenario](#) (string [T](#), [Network](#) [N](#), string [FileName](#))  
*Initializes a new instance of the [NSA.Model.BusinessLogic.Testscenario](#) class.*
- List< [ITestscenarioRunnable](#) > [GetTestscenarioRunnables](#) ()  
*parses the text and creates [Testscenario](#) runnables*

### Propertys

- string [FileName](#) [get]  
*Gets the name of the file.*
- string [Id](#) [get]  
*Gets the identifier.*

### 6.27.1 Beschreibung der Konstruktoren und Destruktoren

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

Initializes a new instance of the [NSA.Model.BusinessLogic.Testscenario](#) class.

#### Parameter

<i>T</i>	Text to parse
<i>N</i>	Network
<i>FileName</i>	File name

### 6.27.2 Dokumentation der Elementfunktionen

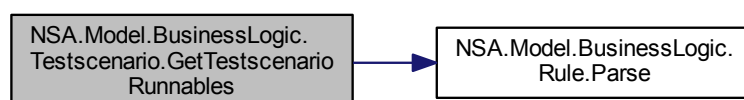
#### 6.27.2.1 List<[ITestscenarioRunnable](#)> NSA.Model.BusinessLogic.Testscenario.GetTestscenarioRunnables ( )

parses the text and creates [Testscenario](#) runnables

#### Rückgabe

runnables

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





### 6.27.3 Dokumentation der Propertys

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

Gets the name of the file.

The name of the file.

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

Gets the identifier.

The identifier.

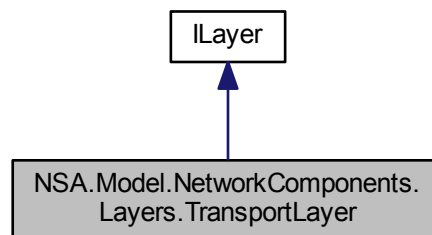
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

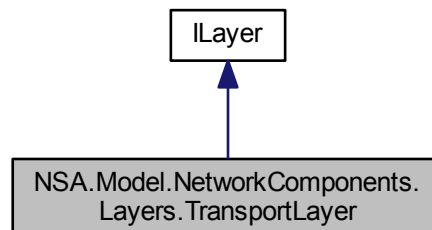
## 6.28 NSA.Model.NetworkComponents.Layers.TransportLayer Klassenreferenz

Transport-Layer

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



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



## Öffentliche Methoden

- [TransportLayer](#) (int I)  
*Initializes a new instance of the [TransportLayer](#) class.*
- bool [ValidateReceive](#) ([Workstation](#) CurrentNode, [ValidationInfo](#) ValInfo, Dictionary< string, object > Tags, [Hardwarenode](#) Destination, int LayerIndex)  
*Validates the layer while receiving a packet.*
- string [GetLayerName](#) ()  
*Gets the name of the layer.*
- bool [SetLayerName](#) (string NewName)  
*Sets the name of the layer.*
- int [GetLayerIndex](#) ()  
*Gets the index of the layer.*
- void [SetLayerIndex](#) (int I)  
*Sets the index of the layer.*
- void [ValidateSend](#) ([Workstation](#) Destination, [Workstation](#) CurrentNode, [ValidationInfo](#) ValInfo, Dictionary< string, object > Tags, int LayerIndex)  
*Validates the layer while sending a packet.*

### 6.28.1 Ausführliche Beschreibung

Transport-Layer

Siehe auch

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

### 6.28.2 Beschreibung der Konstruktoren und Destruktoren

#### 6.28.2.1 NSA.Model.NetworkComponents.Layers.TransportLayer.TransportLayer ( int I )

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

Parameter

I	The i.
---	--------

### 6.28.3 Dokumentation der Elementfunktionen

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

Gets the index of the layer.

Rückgabe

Index

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

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

Gets the name of the layer.

Rückgabe

The Layername

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

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

Sets the index of the layer.

Parameter

I	The Index.
---	------------

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

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

Sets the name of the layer.

Parameter

NewName	New Name
---------	----------

Rückgabe

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

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

Validates the layer while receiving a packet.

Parameter

<i>CurrentNode</i>	Current node
<i>VallInfo</i>	Validation Info
<i>Tags</i>	Tags
<i>Destination</i>	Destinationnode
<i>LayerIndex</i>	Index of the Layer

**Rückgabe**

Boolean value indicating if the validation was successfull

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

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

Validates the layer while sending a packet.

**Parameter**

<i>Destination</i>	The Destination
<i>CurrentNode</i>	Current Node
<i>VallInfo</i>	Validation Info
<i>Tags</i>	Tags
<i>LayerIndex</i>	The Layer index

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

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

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

Helper class for the parameters of the simulation

**Propertys**

- List< [Hardwarenode](#) > [NextNodes](#) [get, set]  
*Gets or sets the next nodes.*
- IPAddress [NextNodeIp](#) [get, set]  
*Gets or sets the next node ip.*
- [Interface](#) [Iface](#) [get, set]  
*Gets or sets the interface.*
- [Result](#) [Res](#) [get, set]  
*Gets or sets the result.*
- [Workstation](#) [Source](#) [get, set]  
*Gets or sets the source.*

### 6.29.1 Ausführliche Beschreibung

Helper class for the parameters of the simulation

## 6.29.2 Dokumentation der Propertys

### 6.29.2.1 Interface NSA.Model.NetworkComponents.Helper\_Classes.ValidationInfo.Iface [get], [set]

Gets or sets the interface.

The interface.

### 6.29.2.2 IPAddress NSA.Model.NetworkComponents.Helper\_Classes.ValidationInfo.NextNodeIp [get], [set]

Gets or sets the next node ip.

The next node ip.

### 6.29.2.3 List<Hardwarenode> NSA.Model.NetworkComponents.Helper\_Classes.ValidationInfo.NextNodes [get], [set]

Gets or sets the next nodes.

The next nodes.

### 6.29.2.4 Result NSA.Model.NetworkComponents.Helper\_Classes.ValidationInfo.Res [get], [set]

Gets or sets the result.

The result.

### 6.29.2.5 Workstation NSA.Model.NetworkComponents.Helper\_Classes.ValidationInfo.Source [get], [set]

Gets or sets the source.

The source.

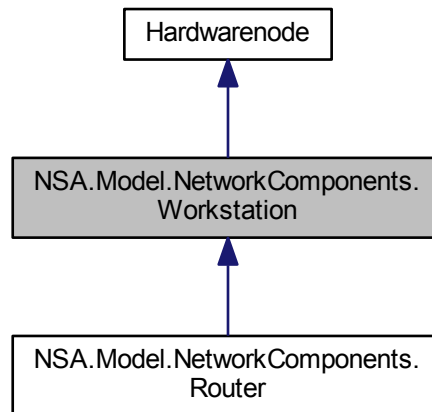
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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

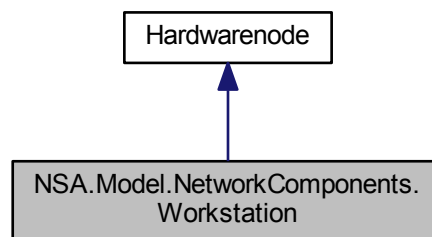
## 6.30 NSA.Model.NetworkComponents.Workstation Klassenreferenz

Implements the network component [Workstation](#).

Klassendiagramm für NSA.Model.NetworkComponents.Workstation:



Zusammengehörigkeiten von NSA.Model.NetworkComponents.Workstation:



### Öffentliche Methoden

- [Workstation](#) (string [Name](#))  
*Initializes a new instance of the [Workstation](#) class. The IP address of the standardgateway must be set seperatly.*
- void [AddRoute](#) ([Route Route](#))  
*Adds the route.*
- void [RemoveRoute](#) (string N)  
*Removes the route.*
- int [GetRouteCount](#) ()

*Gets the route count.*

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

*Gets the routes.*

- [Route GetRouteAt](#) (int Index)

*Gets the route at the given index.*

- bool [SetRoute](#) (string RouteName, IPAddress Destination, IPAddress Subnetmask, IPAddress Gateway, [Interface](#) Iface)

*Sets the route.*

- override bool [HasIp](#) (IPAddress Ip)

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

- override List< [Hardwarenode](#) > [Send](#) ([Hardwarenode](#) Destination, Dictionary< string, object > Tags, [ValidationInfo](#) ValInfo)

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

- override bool [Receive](#) (Dictionary< string, object > Tags, [ValidationInfo](#) ValInfo, [Hardwarenode](#) Destination)

*[Hardwarenode](#) receives the package.*

## Propertys

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

*Gets or sets the standard gateway.*

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

*Gets or sets the standard gateway port.*

## Weitere Geerbte Elemente

### 6.30.1 Ausführliche Beschreibung

Implements the network component [Workstation](#).

Siehe auch

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

### 6.30.2 Beschreibung der Konstruktoren und Destruktoren

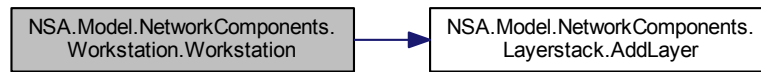
#### 6.30.2.1 NSA.Model.NetworkComponents.Workstation.Workstation ( string Name )

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

Parameter

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

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



### 6.30.3 Dokumentation der Elementfunktionen

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

Adds the route.

Parameter

<i>Route</i>	The route.
--------------	------------

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

Gets the route at the given index.

Parameter

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

Rückgabe

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

Gets the route count.

Rückgabe

int: number of routes in the routingtable

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

Gets the routes.

Rückgabe

The Routes



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

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

Parameter

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

Rückgabe

bool: true if workstation has the ip, otherwise false

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

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

[Hardwarenode](#) receives the package.

Parameter

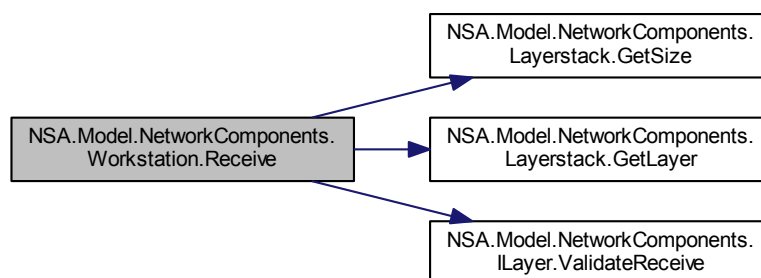
<i>Tags</i>	Optional tags.
<i>VallInfo</i>	The validation Info
<i>Destination</i>	The destination

Rückgabe

bool that indicates if the [Hardwarenode](#) received the package

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

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



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

Removes the route.

## Parameter

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

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

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

## Parameter

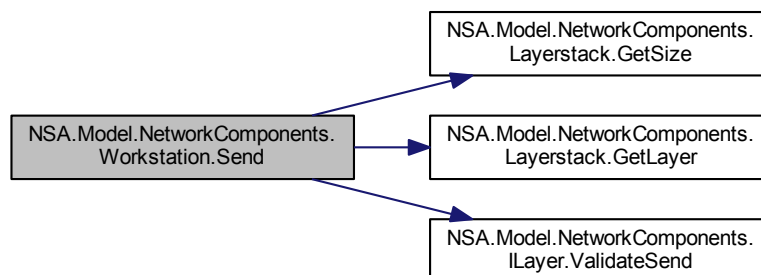
<i>Destination</i>	The destination.
<i>Tags</i>	Optional tags.
<i>ValInfo</i>	

## Rückgabe

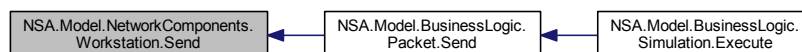
The [Hardwarenode](#) which received the package or null if an error occurred

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

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



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



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

Sets the route.

## Parameter

<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

### 6.30.4 Dokumentation der Propertys

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

Gets or sets the standard gateway.

The standard gateway.

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

Gets or sets the standard gateway port.

The standard gateway port.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

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



## Kapitel 7

# Datei-Dokumentation

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

#### Klassen

- class [NSA.Model.BusinessLogic.Packet](#)  
*Class for a packet*

#### Namensbereiche

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

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

#### Klassen

- class [NSA.Model.BusinessLogic.Project](#)  
*Class for project*

#### Namensbereiche

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

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

#### Klassen

- class [NSA.Model.BusinessLogic.Rule](#)  
*[Rule](#) class parses the text and if it is valid, the [Rule](#) object will be used for simulations*

## Namensbereiche

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

## Aufzählungen

- enum [NSA.Model.BusinessLogic.SimulationType](#) { [NSA.Model.BusinessLogic.SimulationType.Simple](#) = 0, [NSA.Model.BusinessLogic.SimulationType.Only](#) = 1, [NSA.Model.BusinessLogic.SimulationType.HasInternet](#) = 2 }  
*Simulation types*

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

### Klassen

- class [NSA.Model.BusinessLogic.Simulation](#)  
*Class for the simulation*

## Namensbereiche

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

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

### Klassen

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

## Namensbereiche

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

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

### Klassen

- class [NSA.Model.BusinessLogic.TestscenarioRunnables.HasInternetTestscenarioRunnable](#)

#### Namensbereiche

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

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

#### Klassen

- interface [NSA.Model.BusinessLogic.TestscenarioRunnables.ITestscenarioRunnable](#)

#### Namensbereiche

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

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

#### Klassen

- class [NSA.Model.BusinessLogic.TestscenarioRunnables.OnlyTestscenarioRunnable](#)

#### Namensbereiche

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

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

#### Klassen

- class [NSA.Model.BusinessLogic.TestscenarioRunnables.SimpleTestscenarioRunnable](#)

#### Namensbereiche

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

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

### Klassen

- class [NSA.Model.NetworkComponents.Connection](#)  
*Class for a connection between two hardware nodes.*

### Namensbereiche

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

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

### Klassen

- class [NSA.Model.NetworkComponents.Hardwarenode](#)  
*Implements the basis class for hardware nodes.*

### Namensbereiche

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

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

### Klassen

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

### Namensbereiche

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

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

### Klassen

- class [NSA.Model.NetworkComponents.Helper\\_Classes.Result](#)  
*Result class for the packet result*



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

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

### Klassen

- class [NSA.Model.NetworkComponents.Helper\\_Classes.ValidationInfo](#)  
*Helper class for the parameters of the simulation*

### Namensbereiche

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

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

### Klassen

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

### Namensbereiche

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

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

### Klassen

- class [NSA.Model.NetworkComponents.Interface](#)  
*Class for a interface of an hardwarenode.*

### Namensbereiche

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

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

#### Klassen

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

#### Namensbereiche

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

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

#### Klassen

- class [NSA.Model.NetworkComponents.Layers.CustomLayer](#)  
*Custom-Layer*

#### Namensbereiche

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

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

#### Klassen

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

#### Namensbereiche

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

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

#### Klassen

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

## Namensbereiche

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

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

## Klassen

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

## Namensbereiche

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

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

## Klassen

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

## Namensbereiche

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

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

## Klassen

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

## Namensbereiche

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

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

### Klassen

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

### Namensbereiche

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

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

### Klassen

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

### Namensbereiche

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

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

### Klassen

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

### Namensbereiche

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

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

### Klassen

- class [NSA.Model.NetworkComponents.Route](#)  
*class for a single route of the routingtable*

#### Namensbereiche

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

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

#### Klassen

- class [NSA.Model.NetworkComponents.Router](#)  
*Implements the network component [Router](#).*

#### Namensbereiche

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

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

#### Klassen

- class [NSA.Model.NetworkComponents.Switch](#)  
*Implements the network component switch.*

#### Namensbereiche

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

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

#### Klassen

- class [NSA.Model.NetworkComponents.Workstation](#)  
*Implements the network component [Workstation](#).*

#### Namensbereiche

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

- 7.31 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↵  
NSA.Model/obj/Debug/TemporaryGeneratedFile\_036C0B5B-1481-4323-8D20-8F5A↵  
DCB23D92.cs-Dateireferenz
  
- 7.32 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↵  
NSA.Model/obj/Release/TemporaryGeneratedFile\_036C0B5B-1481-4323-8D20-8F5↵  
ADCB23D92.cs-Dateireferenz
  
- 7.33 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↵  
NSA.Model/obj/Debug/TemporaryGeneratedFile\_5937a670-0e60-4077-877b-f7221da3dda1.cs-  
Dateireferenz
  
- 7.34 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↵  
NSA.Model/obj/Release/TemporaryGeneratedFile\_5937a670-0e60-4077-877b-f7221da3dda1.cs-  
Dateireferenz
  
- 7.35 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↵  
NSA.Model/obj/Debug/TemporaryGeneratedFile\_E7A71F73-0F8D-4B9B-B56E-8↵  
E70B10BC5D3.cs-Dateireferenz
  
- 7.36 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↵  
NSA.Model/obj/Release/TemporaryGeneratedFile\_E7A71F73-0F8D-4B9B-B56E-8↵  
E70B10BC5D3.cs-Dateireferenz
  
- 7.37 C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/NetworkSimulatorAnalyzer/↵  
NSA.Model/Properties/AssemblyInfo.cs-Dateireferenz

# Index

AddConnection  
    NSA::Model::NetworkComponents::Hardwarenode, 29  
    NSA::Model::NetworkComponents::Network, 51  
AddHardwarenode  
    NSA::Model::NetworkComponents::Network, 52  
AddInterface  
    NSA::Model::NetworkComponents::Hardwarenode, 29  
    NSA::Model::NetworkComponents::Switch, 90  
AddLayer  
    NSA::Model::NetworkComponents::Layerstack, 46  
AddPacketSend  
    NSA::Model::BusinessLogic::Simulation, 87  
AddRoute  
    NSA::Model::NetworkComponents::Workstation, 102  
ApplicationLayer  
    NSA::Model::NetworkComponents::Layers::↔  
        ApplicationLayer, 14  
  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        BusinessLogic/Packet.cs, 107  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        BusinessLogic/Project.cs, 107  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        BusinessLogic/Rule.cs, 107  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        BusinessLogic/Simulation.cs, 108  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        BusinessLogic/Testscenario.cs, 108  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        BusinessLogic/TestscenarioRunnables/Has↔  
        InternetTestscenarioRunnable.cs, 108  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        BusinessLogic/TestscenarioRunnables/I↔  
        TestscenarioRunnable.cs, 109  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        BusinessLogic/TestscenarioRunnables/↔  
        OnlyTestscenarioRunnable.cs, 109  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
  
BusinessLogic/TestscenarioRunnables/↔  
    SimpleTestscenarioRunnable.cs, 109  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        NetworkComponents/Connection.cs, 110  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        NetworkComponents/Hardwarenode.cs, 110  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        NetworkComponents/Helper    Classes/IP↔  
        AddressExtensions.cs, 110  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        NetworkComponents/Helper    Classes/↔  
        Result.cs, 110  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        NetworkComponents/Helper    Classes/↔  
        ValidationInfo.cs, 111  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        NetworkComponents/ILayer.cs, 111  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        NetworkComponents/Interface.cs, 111  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        NetworkComponents/Layers/Application↔  
        Layer.cs, 112  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        NetworkComponents/Layers/CustomLayer.cs, 112  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        NetworkComponents/Layers/DataLink↔  
        Layer.cs, 112  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        NetworkComponents/Layers/NetworkLayer.↔  
        cs, 112  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        NetworkComponents/Layers/PhysicalLayer.↔  
        cs, 113  
C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔  
    NetworkSimulatorAnalyzer/NSA.Model/↔  
        NetworkComponents/Layers/Presentation↔

- Layer.cs, [113](#)
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
  - NetworkSimulatorAnalyzer/NSA.Model/↔
  - NetworkComponents/Layers/SessionLayer.↔
  - cs, [113](#)
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
  - NetworkSimulatorAnalyzer/NSA.Model/↔
  - NetworkComponents/Layers/Transport↔
  - Layer.cs, [114](#)
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
  - NetworkSimulatorAnalyzer/NSA.Model/↔
  - NetworkComponents/Layerstack.cs, [114](#)
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
  - NetworkSimulatorAnalyzer/NSA.Model/↔
  - NetworkComponents/Network.cs, [114](#)
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
  - NetworkSimulatorAnalyzer/NSA.Model/↔
  - NetworkComponents/Route.cs, [114](#)
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
  - NetworkSimulatorAnalyzer/NSA.Model/↔
  - NetworkComponents/Router.cs, [115](#)
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
  - NetworkSimulatorAnalyzer/NSA.Model/↔
  - NetworkComponents/Switch.cs, [115](#)
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
  - NetworkSimulatorAnalyzer/NSA.Model/↔
  - NetworkComponents/Workstation.cs, [115](#)
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
  - NetworkSimulatorAnalyzer/NSA.Model/↔
  - Properties/AssemblyInfo.cs, [116](#)
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
  - NetworkSimulatorAnalyzer/NSA.Model/obj/↔
  - Debug/TemporaryGeneratedFile\_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs, [116](#)
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
  - NetworkSimulatorAnalyzer/NSA.Model/obj/↔
  - Debug/TemporaryGeneratedFile\_5937a670-0e60-4077-877b-f7221da3dda1.cs, [116](#)
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
  - NetworkSimulatorAnalyzer/NSA.Model/obj/↔
  - Debug/TemporaryGeneratedFile\_E7A71F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs, [116](#)
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
  - NetworkSimulatorAnalyzer/NSA.Model/obj/↔
  - Release/TemporaryGeneratedFile\_036C0↔
  - B5B-1481-4323-8D20-8F5ADCB23D92.cs, [116](#)
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
  - NetworkSimulatorAnalyzer/NSA.Model/obj/↔
  - Release/TemporaryGeneratedFile\_5937a670-0e60-4077-877b-f7221da3dda1.cs, [116](#)
- C:/SWP16/Basisverzeichnis/trunk/03\_Implementierung/↔
  - NetworkSimulatorAnalyzer/NSA.Model/obj/↔
  - Release/TemporaryGeneratedFile\_E7A71↔
  - F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs, [116](#)
- CheckForTrueOrFalse
  - NSA::Model::BusinessLogic::Rule, [79](#)
- Connection
  - NSA::Model::NetworkComponents::Connection, [17](#)
- Connections
  - NSA::Model::NetworkComponents::Hardwarenode, [35](#)
  - NSA::Model::NetworkComponents::Network, [54](#)
- CreateUniqueName
  - NSA::Model::NetworkComponents::Layerstack, [47](#)
- CustomLayer
  - NSA::Model::NetworkComponents::Layers::↔
  - CustomLayer, [21](#)
- CustomLayerError
  - NSA::Model::NetworkComponents::Helper\_↔
  - Classes::Result, [71](#)
- CustomLayerIndexError
  - NSA::Model::NetworkComponents::Helper\_↔
  - Classes::Result, [71](#)
- DataLinkLayer
  - NSA::Model::NetworkComponents::Layers::Data↔
  - LinkLayer, [25](#)
- Destination
  - NSA::Model::BusinessLogic::Packet, [61](#)
  - NSA::Model::BusinessLogic::Simulation, [88](#)
  - NSA::Model::NetworkComponents::Route, [75](#)
- End
  - NSA::Model::NetworkComponents::Connection, [19](#)
- EndNodes
  - NSA::Model::BusinessLogic::Rule, [80](#)
- EndNodesString
  - NSA::Model::BusinessLogic::Rule, [80](#)
- Equals
  - NSA::Model::NetworkComponents::Connection, [17, 18](#)
  - NSA::Model::NetworkComponents::Hardwarenode, [30](#)
- ErrorId
  - NSA::Model::NetworkComponents::Helper\_↔
  - Classes::Result, [73](#)
- Errors
  - NSA::Model::NetworkComponents::Helper\_↔
  - Classes::Result, [71](#)
- Execute
  - NSA::Model::BusinessLogic::Simulation, [87](#)
- ExpectedResult
  - NSA::Model::BusinessLogic::Packet, [61](#)
  - NSA::Model::BusinessLogic::Rule, [80](#)
  - NSA::Model::BusinessLogic::Simulation, [88](#)
- FileName
  - NSA::Model::BusinessLogic::Testscenario, [95](#)
- Gateway
  - NSA::Model::NetworkComponents::Route, [75](#)
- GetAllConnections
  - NSA::Model::NetworkComponents::Network, [52](#)
- GetAllHardwarenodes
  - NSA::Model::NetworkComponents::Network, [52](#)



- GetAllLayers
  - NSA::Model::NetworkComponents::Layerstack, [47](#)
- GetAllPackets
  - NSA::Model::BusinessLogic::Simulation, [87](#)
- GetAllWorkstations
  - NSA::Model::NetworkComponents::Network, [52](#)
- GetBroadcastAddress
  - NSA::Model::NetworkComponents::Helper\_↔
    - Classes::IPAddressExtensions, [44](#)
- GetConnectionAtPort
  - NSA::Model::NetworkComponents::Hardwarenode, [31](#)
- GetConnectionByName
  - NSA::Model::NetworkComponents::Network, [52](#)
- GetHardwareNodesForSubnet
  - NSA::Model::NetworkComponents::Network, [53](#)
- GetHardwarenodeByName
  - NSA::Model::NetworkComponents::Network, [53](#)
- GetHashCode
  - NSA::Model::NetworkComponents::Connection, [18](#)
  - NSA::Model::NetworkComponents::Hardwarenode, [31](#)
- GetInterfaceCount
  - NSA::Model::NetworkComponents::Hardwarenode, [31](#)
- GetLastPacket
  - NSA::Model::BusinessLogic::Simulation, [87](#)
- GetLayer
  - NSA::Model::NetworkComponents::Layerstack, [47](#)
- GetLayerByName
  - NSA::Model::NetworkComponents::Layerstack, [47](#)
- GetLayerIndex
  - NSA::Model::NetworkComponents::ILayer, [39](#)
  - NSA::Model::NetworkComponents::Layers::↔
    - ApplicationLayer, [14](#)
  - NSA::Model::NetworkComponents::Layers::↔
    - CustomLayer, [21](#)
  - NSA::Model::NetworkComponents::Layers::Data↔
    - LinkLayer, [25](#)
  - NSA::Model::NetworkComponents::Layers::↔
    - NetworkLayer, [56](#)
  - NSA::Model::NetworkComponents::Layers::↔
    - PhysicalLayer, [64](#)
  - NSA::Model::NetworkComponents::Layers::↔
    - PresentationLayer, [67](#)
  - NSA::Model::NetworkComponents::Layers::↔
    - SessionLayer, [82](#)
  - NSA::Model::NetworkComponents::Layers::↔
    - TransportLayer, [96](#)
- GetLayerName
  - NSA::Model::NetworkComponents::ILayer, [39](#)
  - NSA::Model::NetworkComponents::Layers::↔
    - ApplicationLayer, [14](#)
  - NSA::Model::NetworkComponents::Layers::↔
    - CustomLayer, [21](#)
  - NSA::Model::NetworkComponents::Layers::Data↔
    - LinkLayer, [25](#)
  - NSA::Model::NetworkComponents::Layers::↔
- NetworkLayer, [56](#)
- NSA::Model::NetworkComponents::Layers::↔
  - PhysicalLayer, [64](#)
- NSA::Model::NetworkComponents::Layers::↔
  - PresentationLayer, [67](#)
- NSA::Model::NetworkComponents::Layers::↔
  - SessionLayer, [82](#)
- NSA::Model::NetworkComponents::Layers::↔
  - TransportLayer, [96](#)
- GetNetworkAddress
  - NSA::Model::NetworkComponents::Helper\_↔
    - Classes::IPAddressExtensions, [44](#)
- getNewInterfaceNumber
  - NSA::Model::NetworkComponents::Hardwarenode, [31](#)
- GetPortIndex
  - NSA::Model::NetworkComponents::Connection, [18](#)
- GetPortIndexOfConnection
  - NSA::Model::NetworkComponents::Hardwarenode, [31](#)
- GetRouteAt
  - NSA::Model::NetworkComponents::Workstation, [102](#)
- GetRouteCount
  - NSA::Model::NetworkComponents::Workstation, [102](#)
- GetRouters
  - NSA::Model::NetworkComponents::Network, [53](#)
- GetRoutes
  - NSA::Model::NetworkComponents::Workstation, [102](#)
- GetSize
  - NSA::Model::NetworkComponents::Layerstack, [48](#)
- GetTestscenarioRunnables
  - NSA::Model::BusinessLogic::Testscenario, [94](#)
- GetWorkstationByIp
  - NSA::Model::NetworkComponents::Network, [53](#)
- Hardwarenode
  - NSA::Model::NetworkComponents::Hardwarenode, [29](#)
- HasInterface
  - NSA::Model::NetworkComponents::Hardwarenode, [32](#)
- HasInternet
  - NSA::Model::BusinessLogic, [10](#)
- HasInternetTestscenarioRunnable
  - NSA::Model::BusinessLogic::Testscenario↔
    - Runnables::HasInternetTestscenario↔
      - Runnable, [37](#)
- HasIp
  - NSA::Model::NetworkComponents::Hardwarenode, [32](#)
  - NSA::Model::NetworkComponents::Workstation, [102](#)
- Hops
  - NSA::Model::BusinessLogic::Packet, [62](#)
- Id

- NSA::Model::BusinessLogic::Simulation, 88
- NSA::Model::BusinessLogic::Testscenario, 95
- iface
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::ValidationInfo, 99
  - NSA::Model::NetworkComponents::Route, 75
- InsertAt
  - NSA::Model::NetworkComponents::Layerstack, 48
- Interface
  - NSA::Model::NetworkComponents::Interface, 43
- InterfacelsUsed
  - NSA::Model::NetworkComponents::Hardwarenode, 33
- Interfaces
  - NSA::Model::NetworkComponents::Hardwarenode, 35
- IpAddress
  - NSA::Model::NetworkComponents::Interface, 43
- IsGateway
  - NSA::Model::NetworkComponents::Router, 77
- IsInSameSubnet
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::IPAddressExtensions, 44
- IsNameTaken
  - NSA::Model::NetworkComponents::Layerstack, 48
- IsValidSubnetMask
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::IPAddressExtensions, 44
- LayerError
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::Result, 73
- Layerstack
  - NSA::Model::NetworkComponents::Hardwarenode, 35
  - NSA::Model::NetworkComponents::Layerstack, 46
- NSA.Model, 9
- NSA.Model.BusinessLogic, 9
- NSA.Model.BusinessLogic.Packet, 60
- NSA.Model.BusinessLogic.Project, 69
- NSA.Model.BusinessLogic.Rule, 77
- NSA.Model.BusinessLogic.Simulation, 86
- NSA.Model.BusinessLogic.Testscenario, 94
- NSA.Model.BusinessLogic.TestscenarioRunnables, 10
- NSA.Model.BusinessLogic.TestscenarioRunnables.↔
  - HasInternetTestscenarioRunnable, 36
- NSA.Model.BusinessLogic.TestscenarioRunnables.I↔
  - TestscenarioRunnable, 45
- NSA.Model.BusinessLogic.TestscenarioRunnables.↔
  - OnlyTestscenarioRunnable, 58
- NSA.Model.BusinessLogic.TestscenarioRunnables.↔
  - SimpleTestscenarioRunnable, 84
- NSA.Model.NetworkComponents, 10
- NSA.Model.NetworkComponents.Connection, 16
- NSA.Model.NetworkComponents.Hardwarenode, 27
- NSA.Model.NetworkComponents.Helper\_Classes, 11
- NSA.Model.NetworkComponents.Helper\_Classes.IP↔
  - AddressExtensions, 44
- NSA.Model.NetworkComponents.Helper\_Classes.↔
  - Result, 70
- NSA.Model.NetworkComponents.Helper\_Classes.↔
  - ValidationInfo, 98
- NSA.Model.NetworkComponents.ILayer, 38
- NSA.Model.NetworkComponents.Interface, 42
- NSA.Model.NetworkComponents.Layers, 11
- NSA.Model.NetworkComponents.Layers.Application↔
  - Layer, 13
- NSA.Model.NetworkComponents.Layers.CustomLayer, 20
- NSA.Model.NetworkComponents.Layers.DataLink↔
  - Layer, 24
- NSA.Model.NetworkComponents.Layers.NetworkLayer, 54
- NSA.Model.NetworkComponents.Layers.PhysicalLayer, 63
- NSA.Model.NetworkComponents.Layers.Presentation↔
  - Layer, 66
- NSA.Model.NetworkComponents.Layers.SessionLayer, 81
- NSA.Model.NetworkComponents.Layers.Transport↔
  - Layer, 95
- NSA.Model.NetworkComponents.Layerstack, 45
- NSA.Model.NetworkComponents.Network, 51
- NSA.Model.NetworkComponents.Route, 73
- NSA.Model.NetworkComponents.Router, 76
- NSA.Model.NetworkComponents.Switch, 89
- NSA.Model.NetworkComponents.Workstation, 100
- NSA::Model::BusinessLogic
  - HasInternet, 10
  - Only, 10
  - Simple, 10
  - SimulationType, 10
- NSA::Model::BusinessLogic::Packet
  - Destination, 61
  - ExpectedResult, 61
  - Hops, 62
  - Packet, 60
  - Result, 62
  - Send, 61
  - Source, 62
  - Ttl, 62
- NSA::Model::BusinessLogic::Project
  - Network, 70
  - Path, 70
  - Project, 70
- NSA::Model::BusinessLogic::Rule
  - CheckForTrueOrFalse, 79
  - EndNodes, 80
  - EndNodesString, 80
  - ExpectedResult, 80
  - Options, 80
  - Parameters, 80
  - Parse, 79
  - Rule, 78
  - SimulType, 80
  - StartNode, 80

- StartNodeString, 80
- NSA::Model::BusinessLogic::Simulation
  - AddPacketSend, 87
  - Destination, 88
  - Execute, 87
  - ExpectedResult, 88
  - GetAllPackets, 87
  - GetLastPacket, 87
  - Id, 88
  - PacketsReceived, 88
  - PacketsSend, 88
  - Simulation, 86, 87
  - Source, 88
- NSA::Model::BusinessLogic::Testscenario
  - FileName, 95
  - GetTestscenarioRunnables, 94
  - Id, 95
  - Testscenario, 94
- NSA::Model::BusinessLogic::TestscenarioRunnables↔
  - ::HasInternetTestscenarioRunnable
  - HasInternetTestscenarioRunnable, 37
  - Run, 37
- NSA::Model::BusinessLogic::TestscenarioRunnables↔
  - ::ITestscenarioRunnable
  - Run, 45
- NSA::Model::BusinessLogic::TestscenarioRunnables↔
  - ::OnlyTestscenarioRunnable
  - OnlyTestscenarioRunnable, 59
  - Run, 59
- NSA::Model::BusinessLogic::TestscenarioRunnables↔
  - ::SimpleTestscenarioRunnable
  - Run, 85
  - SimpleTestscenarioRunnable, 85
- NSA::Model::NetworkComponents::Connection
  - Connection, 17
  - End, 19
  - Equals, 17, 18
  - GetHashCode, 18
  - GetPortIndex, 18
  - Name, 19
  - operator!=, 18
  - operator==, 19
  - Start, 19
- NSA::Model::NetworkComponents::Hardwarenode
  - AddConnection, 29
  - AddInterface, 29
  - Connections, 35
  - Equals, 30
  - GetConnectionAtPort, 31
  - GetHashCode, 31
  - GetInterfaceCount, 31
  - getNewInterfaceNumber, 31
  - GetPortIndexOfConnection, 31
  - Hardwarenode, 29
  - HasInterface, 32
  - HasIp, 32
  - InterfacelsUsed, 33
  - Interfaces, 35
- Layerstack, 35
- Name, 36
- operator!=, 33
- operator==, 33
- Receive, 34
- RemoveConnection, 34
- RemoveInterface, 34
- Send, 35
- SetInterface, 35
- NSA::Model::NetworkComponents::Helper\_Classes::I↔
  - PAddressExtensions
  - GetBroadcastAddress, 44
  - GetNetworkAddress, 44
  - IsInSameSubnet, 44
  - IsValidSubnetMask, 44
- NSA::Model::NetworkComponents::Helper\_Classes::I↔
  - Result
  - CustomLayerError, 71
  - CustomLayerIndexError, 71
  - ErrorId, 73
  - Errors, 71
  - LayerError, 73
  - NoConnection, 71
  - NoError, 71
  - NoPackets, 71
  - NoRoute, 71
  - PacketNotForThisNode, 71
  - Res, 73
  - Result, 72
  - ResultStrings, 72
  - SendError, 73
  - SourceDestinationNull, 71
  - SwitchNoConnection, 71
  - TtlError, 71
- NSA::Model::NetworkComponents::Helper\_Classes::I↔
  - ValidationInfo
  - Iface, 99
  - NextNodeIp, 99
  - NextNodes, 99
  - Res, 99
  - Source, 99
- NSA::Model::NetworkComponents::ILayer
  - GetLayerIndex, 39
  - GetLayerName, 39
  - SetLayerIndex, 39
  - SetLayerName, 40
  - ValidateReceive, 41
  - ValidateSend, 41
- NSA::Model::NetworkComponents::Interface
  - Interface, 43
  - IpAddress, 43
  - Name, 43
  - NamePrefix, 43
  - SetInterface, 43
  - Subnetmask, 44
- NSA::Model::NetworkComponents::Layers::Application↔
  - Layer
  - ApplicationLayer, 14

- GetLayerIndex, [14](#)
- GetLayerName, [14](#)
- SetLayerIndex, [15](#)
- SetLayerName, [15](#)
- ValidateReceive, [15](#)
- ValidateSend, [16](#)
- NSA::Model::NetworkComponents::Layers::Custom↔
  - Layer
  - CustomLayer, [21](#)
  - GetLayerIndex, [21](#)
  - GetLayerName, [21](#)
  - SetLayerIndex, [22](#)
  - SetLayerName, [22](#)
  - ValidateReceive, [22](#)
  - ValidateSend, [23](#)
- NSA::Model::NetworkComponents::Layers::DataLink↔
  - Layer
  - DataLinkLayer, [25](#)
  - GetLayerIndex, [25](#)
  - GetLayerName, [25](#)
  - SetLayerIndex, [25](#)
  - SetLayerName, [26](#)
  - ValidateReceive, [26](#)
  - ValidateSend, [26](#)
- NSA::Model::NetworkComponents::Layers::Network↔
  - Layer
  - GetLayerIndex, [56](#)
  - GetLayerName, [56](#)
  - NetworkLayer, [56](#)
  - SetLayerIndex, [56](#)
  - SetLayerName, [57](#)
  - ValidateReceive, [57](#)
  - ValidateSend, [57](#)
- NSA::Model::NetworkComponents::Layers::Physical↔
  - Layer
  - GetLayerIndex, [64](#)
  - GetLayerName, [64](#)
  - PhysicalLayer, [64](#)
  - SetLayerIndex, [64](#)
  - SetLayerName, [65](#)
  - ValidateReceive, [65](#)
  - ValidateSend, [65](#)
- NSA::Model::NetworkComponents::Layers::Presentation↔
  - Layer
  - GetLayerIndex, [67](#)
  - GetLayerName, [67](#)
  - PresentationLayer, [67](#)
  - SetLayerIndex, [68](#)
  - SetLayerName, [68](#)
  - ValidateReceive, [68](#)
  - ValidateSend, [69](#)
- NSA::Model::NetworkComponents::Layers::Session↔
  - Layer
  - GetLayerIndex, [82](#)
  - GetLayerName, [82](#)
  - SessionLayer, [82](#)
  - SetLayerIndex, [83](#)
  - SetLayerName, [83](#)
- ValidateReceive, [83](#)
- ValidateSend, [84](#)
- NSA::Model::NetworkComponents::Layers::Transport↔
  - Layer
  - GetLayerIndex, [96](#)
  - GetLayerName, [96](#)
  - SetLayerIndex, [97](#)
  - SetLayerName, [97](#)
  - TransportLayer, [96](#)
  - ValidateReceive, [97](#)
  - ValidateSend, [98](#)
- NSA::Model::NetworkComponents::Layerstack
  - AddLayer, [46](#)
  - CreateUniqueName, [47](#)
  - GetAllLayers, [47](#)
  - GetLayer, [47](#)
  - GetLayerByName, [47](#)
  - GetSize, [48](#)
  - InsertAt, [48](#)
  - IsNameTaken, [48](#)
  - Layerstack, [46](#)
  - RemoveLayer, [49](#)
  - SetIndex, [49](#)
  - SetName, [50](#)
- NSA::Model::NetworkComponents::Network
  - AddConnection, [51](#)
  - AddHardwarenode, [52](#)
  - Connections, [54](#)
  - GetAllConnections, [52](#)
  - GetAllHardwarenodes, [52](#)
  - GetAllWorkstations, [52](#)
  - GetConnectionByName, [52](#)
  - GetHardwareNodesForSubnet, [53](#)
  - GetHardwarenodeByName, [53](#)
  - GetRouters, [53](#)
  - GetWorkstationByIp, [53](#)
  - Network, [51](#)
  - RemoveConnection, [54](#)
  - RemoveHardwarenode, [54](#)
- NSA::Model::NetworkComponents::Route
  - Destination, [75](#)
  - Gateway, [75](#)
  - Iface, [75](#)
  - Name, [75](#)
  - Route, [74](#)
  - SetRoute, [74](#)
  - Subnetmask, [75](#)
- NSA::Model::NetworkComponents::Router
  - IsGateway, [77](#)
  - Router, [77](#)
- NSA::Model::NetworkComponents::Switch
  - AddInterface, [90](#)
  - Send, [90](#)
  - SendToDestination, [91](#)
  - SendToIp, [92](#)
  - SetInterfaceCount, [93](#)
  - Switch, [90](#)
- NSA::Model::NetworkComponents::Workstation

- AddRoute, [102](#)
- GetRouteAt, [102](#)
- GetRouteCount, [102](#)
- GetRoutes, [102](#)
- HasIp, [102](#)
- Receive, [103](#)
- RemoveRoute, [103](#)
- Send, [104](#)
- SetRoute, [104](#)
- StandardGateway, [105](#)
- StandardGatewayPort, [105](#)
- Workstation, [101](#)
- NSA, [9](#)
- Name
  - NSA::Model::NetworkComponents::Connection, [19](#)
  - NSA::Model::NetworkComponents::Hardwarenode, [36](#)
  - NSA::Model::NetworkComponents::Interface, [43](#)
  - NSA::Model::NetworkComponents::Route, [75](#)
- NamePrefix
  - NSA::Model::NetworkComponents::Interface, [43](#)
- Network
  - NSA::Model::BusinessLogic::Project, [70](#)
  - NSA::Model::NetworkComponents::Network, [51](#)
- NetworkLayer
  - NSA::Model::NetworkComponents::Layers::↔  
NetworkLayer, [56](#)
- NextNodeIp
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::ValidationInfo, [99](#)
- NextNodes
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::ValidationInfo, [99](#)
- NoConnection
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::Result, [71](#)
- NoError
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::Result, [71](#)
- NoPackets
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::Result, [71](#)
- NoRoute
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::Result, [71](#)
- Only
  - NSA::Model::BusinessLogic, [10](#)
- OnlyTestscenarioRunnable
  - NSA::Model::BusinessLogic::Testscenario↔  
Runnables::OnlyTestscenarioRunnable, [59](#)
- operator!=
  - NSA::Model::NetworkComponents::Connection, [18](#)
  - NSA::Model::NetworkComponents::Hardwarenode, [33](#)
- operator==
  - NSA::Model::NetworkComponents::Connection, [19](#)
  - NSA::Model::NetworkComponents::Hardwarenode, [33](#)
- Options
  - NSA::Model::BusinessLogic::Rule, [80](#)
- Packet
  - NSA::Model::BusinessLogic::Packet, [60](#)
- PacketNotForThisNode
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::Result, [71](#)
- PacketsReceived
  - NSA::Model::BusinessLogic::Simulation, [88](#)
- PacketsSend
  - NSA::Model::BusinessLogic::Simulation, [88](#)
- Parameters
  - NSA::Model::BusinessLogic::Rule, [80](#)
- Parse
  - NSA::Model::BusinessLogic::Rule, [79](#)
- Path
  - NSA::Model::BusinessLogic::Project, [70](#)
- PhysicalLayer
  - NSA::Model::NetworkComponents::Layers::↔  
PhysicalLayer, [64](#)
- PresentationLayer
  - NSA::Model::NetworkComponents::Layers::↔  
PresentationLayer, [67](#)
- Project
  - NSA::Model::BusinessLogic::Project, [70](#)
- Receive
  - NSA::Model::NetworkComponents::Hardwarenode, [34](#)
  - NSA::Model::NetworkComponents::Workstation, [103](#)
- RemoveConnection
  - NSA::Model::NetworkComponents::Hardwarenode, [34](#)
  - NSA::Model::NetworkComponents::Network, [54](#)
- RemoveHardwarnode
  - NSA::Model::NetworkComponents::Network, [54](#)
- RemoveInterface
  - NSA::Model::NetworkComponents::Hardwarenode, [34](#)
- RemoveLayer
  - NSA::Model::NetworkComponents::Layerstack, [49](#)
- RemoveRoute
  - NSA::Model::NetworkComponents::Workstation, [103](#)
- Res
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::Result, [73](#)
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::ValidationInfo, [99](#)
- Result
  - NSA::Model::BusinessLogic::Packet, [62](#)
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::Result, [72](#)
- ResultStrings
  - NSA::Model::NetworkComponents::Helper\_↔  
Classes::Result, [72](#)
- Route

- NSA::Model::NetworkComponents::Route, 74
- Router
  - NSA::Model::NetworkComponents::Router, 77
- Rule
  - NSA::Model::BusinessLogic::Rule, 78
- Run
  - NSA::Model::BusinessLogic::Testscenario↔
    - Runnables::HasInternetTestscenario↔
      - Runnable, 37
  - NSA::Model::BusinessLogic::Testscenario↔
    - Runnables::ITestscenarioRunnable, 45
  - NSA::Model::BusinessLogic::Testscenario↔
    - Runnables::OnlyTestscenarioRunnable, 59
  - NSA::Model::BusinessLogic::Testscenario↔
    - Runnables::SimpleTestscenarioRunnable, 85
- Send
  - NSA::Model::BusinessLogic::Packet, 61
  - NSA::Model::NetworkComponents::Hardwarenode, 35
  - NSA::Model::NetworkComponents::Switch, 90
  - NSA::Model::NetworkComponents::Workstation, 104
- SendError
  - NSA::Model::NetworkComponents::Helper\_↔
    - Classes::Result, 73
- SendToDestination
  - NSA::Model::NetworkComponents::Switch, 91
- SendToIp
  - NSA::Model::NetworkComponents::Switch, 92
- SessionLayer
  - NSA::Model::NetworkComponents::Layers::↔
    - SessionLayer, 82
- SetIndex
  - NSA::Model::NetworkComponents::Layerstack, 49
- SetInterface
  - NSA::Model::NetworkComponents::Hardwarenode, 35
  - NSA::Model::NetworkComponents::Interface, 43
- SetInterfaceCount
  - NSA::Model::NetworkComponents::Switch, 93
- SetLayerIndex
  - NSA::Model::NetworkComponents::ILayer, 39
  - NSA::Model::NetworkComponents::Layers::↔
    - ApplicationLayer, 15
  - NSA::Model::NetworkComponents::Layers::↔
    - CustomLayer, 22
  - NSA::Model::NetworkComponents::Layers::Data↔
    - LinkLayer, 25
  - NSA::Model::NetworkComponents::Layers::↔
    - NetworkLayer, 56
  - NSA::Model::NetworkComponents::Layers::↔
    - PhysicalLayer, 64
  - NSA::Model::NetworkComponents::Layers::↔
    - PresentationLayer, 68
  - NSA::Model::NetworkComponents::Layers::↔
    - SessionLayer, 83
  - NSA::Model::NetworkComponents::Layers::↔
    - TransportLayer, 97
- SetLayerName
  - NSA::Model::NetworkComponents::ILayer, 40
  - NSA::Model::NetworkComponents::Layers::↔
    - ApplicationLayer, 15
  - NSA::Model::NetworkComponents::Layers::↔
    - CustomLayer, 22
  - NSA::Model::NetworkComponents::Layers::Data↔
    - LinkLayer, 26
  - NSA::Model::NetworkComponents::Layers::↔
    - NetworkLayer, 57
  - NSA::Model::NetworkComponents::Layers::↔
    - PhysicalLayer, 65
  - NSA::Model::NetworkComponents::Layers::↔
    - PresentationLayer, 68
  - NSA::Model::NetworkComponents::Layers::↔
    - SessionLayer, 83
  - NSA::Model::NetworkComponents::Layers::↔
    - TransportLayer, 97
- SetName
  - NSA::Model::NetworkComponents::Layerstack, 50
- SetRoute
  - NSA::Model::NetworkComponents::Route, 74
  - NSA::Model::NetworkComponents::Workstation, 104
- Simple
  - NSA::Model::BusinessLogic, 10
- SimpleTestscenarioRunnable
  - NSA::Model::BusinessLogic::Testscenario↔
    - Runnables::SimpleTestscenarioRunnable, 85
- SimulType
  - NSA::Model::BusinessLogic::Rule, 80
- Simulation
  - NSA::Model::BusinessLogic::Simulation, 86, 87
- SimulationType
  - NSA::Model::BusinessLogic, 10
- Source
  - NSA::Model::BusinessLogic::Packet, 62
  - NSA::Model::BusinessLogic::Simulation, 88
  - NSA::Model::NetworkComponents::Helper\_↔
    - Classes::ValidationInfo, 99
- SourceDestinationNull
  - NSA::Model::NetworkComponents::Helper\_↔
    - Classes::Result, 71
- StandardGateway
  - NSA::Model::NetworkComponents::Workstation, 105
- StandardGatewayPort
  - NSA::Model::NetworkComponents::Workstation, 105
- Start
  - NSA::Model::NetworkComponents::Connection, 19
- StartNode
  - NSA::Model::BusinessLogic::Rule, 80
- StartNodeString
  - NSA::Model::BusinessLogic::Rule, 80
- Subnetmask
  - NSA::Model::NetworkComponents::Interface, 44
  - NSA::Model::NetworkComponents::Route, 75



## Switch

NSA::Model::NetworkComponents::Switch, [90](#)

## SwitchNoConnection

NSA::Model::NetworkComponents::Helper\_↔  
Classes::Result, [71](#)

## Testscenario

NSA::Model::BusinessLogic::Testscenario, [94](#)

## TransportLayer

NSA::Model::NetworkComponents::Layers::↔  
TransportLayer, [96](#)

## Ttl

NSA::Model::BusinessLogic::Packet, [62](#)

## TtlError

NSA::Model::NetworkComponents::Helper\_↔  
Classes::Result, [71](#)

## ValidateReceive

NSA::Model::NetworkComponents::ILayer, [41](#)  
NSA::Model::NetworkComponents::Layers::↔  
ApplicationLayer, [15](#)  
NSA::Model::NetworkComponents::Layers::↔  
CustomLayer, [22](#)  
NSA::Model::NetworkComponents::Layers::Data↔  
LinkLayer, [26](#)  
NSA::Model::NetworkComponents::Layers::↔  
NetworkLayer, [57](#)  
NSA::Model::NetworkComponents::Layers::↔  
PhysicalLayer, [65](#)  
NSA::Model::NetworkComponents::Layers::↔  
PresentationLayer, [68](#)  
NSA::Model::NetworkComponents::Layers::↔  
SessionLayer, [83](#)  
NSA::Model::NetworkComponents::Layers::↔  
TransportLayer, [97](#)

## ValidateSend

NSA::Model::NetworkComponents::ILayer, [41](#)  
NSA::Model::NetworkComponents::Layers::↔  
ApplicationLayer, [16](#)  
NSA::Model::NetworkComponents::Layers::↔  
CustomLayer, [23](#)  
NSA::Model::NetworkComponents::Layers::Data↔  
LinkLayer, [26](#)  
NSA::Model::NetworkComponents::Layers::↔  
NetworkLayer, [57](#)  
NSA::Model::NetworkComponents::Layers::↔  
PhysicalLayer, [65](#)  
NSA::Model::NetworkComponents::Layers::↔  
PresentationLayer, [69](#)  
NSA::Model::NetworkComponents::Layers::↔  
SessionLayer, [84](#)  
NSA::Model::NetworkComponents::Layers::↔  
TransportLayer, [98](#)

## Workstation

NSA::Model::NetworkComponents::Workstation,  
[101](#)