All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package com.meshdynamics.api

Interface Summary

NMS.ConnectedDevice	Defines the properties of all devices connected to a `NMS.Node`
NMS.NeighborNode	Defines the properties of all neighbor nodes detected by a `NMS.Node`
NMS.Network	The `Network` interface defines all properties and actions associated with a mesh network.
NMS.NetworkListener	The `NetworkListener` interface is used to receive events on a mesh network.
NMS.Node	The `Node` interface defines all the properties and actions that can be carried out on a mesh node.
NMS.Thread.Runnable	The `Runnable` interface is implemented by any class whose instances are executed by a thread.

Class Summary

NMS	NMS is the primary class for using the **Meshdynamics Network Management System (NMS) API**.
NMS.ACLConfiguration	Defines the Access Control List configuration for a node.
NMS.ACLEntry	Defines an Access Control List entry.
NMS.EffistreamRule	Defines a Effistream QoS rule.
NMS.GeneralConfiguration	Defines all Node level fields used by a `NMS.Node`.
NMS.Hashtable	The Hashtable class provides an implementation of a Hashtable of generic 'Object' keys and generic 'Object' values.
NMS.InterfaceConfiguration	Defines the interface level settings for a `NMS.Node`.
NMS.ObjectArray	The ObjectArray class provides an interface to a growable array that stores object references.
NMS.ShortArray	Defines an array of short integers.
NMS.Thread	The `Thread` class provides multi-threading functionality to scripting platforms.
NMS.VlanConfiguration	Defines the settings for a Virtual-LAN in a `NMS.Node`.
NMS.WEPSecurity	Defines the information used by the IEEE 802.11 **Wired Equivalent Privacy** (WEP) setting by a Node's downlink interface.

NMS.WPAEnterpriseSecurity	Defines the information used for the Wifi Protected Access security setting by a Node's downlink interface in an enterprise environment.
NMS.WPAPersonalSecurity	Defines the information used for the Wifi Protected Access (WPA) security setting by a node's downlink interface.

Package Class **Tree** **Deprecated** **Index** **Help**

PREV PACKAGE NEXT PACKAGE **FRAMES** **NO FRAMES**

com.meshdynamics.api

Class NMS

```
java.lang.Object
  └ com.meshdynamics.api.NMS
```

```
public abstract class NMS
extends java.lang.Object
```

NMS is the primary class for using the **Meshdynamics Network Management System (NMS) API**.

It is a singleton class defining classes, interfaces and constants to be used for accessing the NMS information

All clients of the NMS API need to obtain a reference to the singleton instance of the NMS object by calling the `NMS.getInstance()` method.

Nested Class Summary

static class	**NMS.ACLConfiguration** Defines the Access Control List configuration for a node.
static class	**NMS.ACLEntry** Defines an Access Control List entry.
static interface	**NMS.ConnectedDevice** Defines the properties of all devices connected to a NMS.Node
static class	**NMS.EffistreamRule** Defines a Effistream QoS rule.
static class	**NMS.GeneralConfiguration** Defines all Node level fields used by a NMS.Node.
static class	**NMS.Hashtable** The Hashtable class provides an implementation of a Hashtable of generic 'Object' keys and generic 'Object' values.
static class	**NMS.InterfaceConfiguration** Defines the interface level settings for a NMS.Node.
static interface	**NMS.NeighborNode** Defines the properties of all neighbor nodes detected by a NMS.Node
static interface	**NMS.Network** The Network interface defines all properties and actions associated with a mesh network.
static interface	**NMS.NetworkListener** The NetworkListener interface is used to receive events on a mesh network.
static interface	**NMS.Node** The Node interface defines all the properties and actions that can be carried out on a mesh node.

static class	**NMS.ObjectArray** The ObjectArray class provides an interface to a growable array that stores object references.
static class	**NMS.ShortArray** Defines an array of short integers.
static class	**NMS.Thread** The Thread class provides multi-threading functionality to scripting platforms.
static class	**NMS.VlanConfiguration** Defines the settings for a Virtual-LAN in a NMS.Node.
static class	**NMS.WEPSecurity** Defines the information used by the IEEE 802.11 **Wired Equivalent Privacy** (WEP) setting by a Node's downlink interface.
static class	**NMS.WPAEnterpriseSecurity** Defines the information used for the Wifi Protected Access security setting by a Node's downlink interface in an enterprise environment.
static class	**NMS.WPAPersonalSecurity** Defines the information used for the Wifi Protected Access (WPA) security setting by a node's downlink interface.

Field Summary

static short	**CIPHER_CCMP** Specifies AES-CCMP encryption to be used for WPA/WPA2 Personal/Enterprise security options.
static short	**CIPHER_TKIP** Specifies TKIP encryption to be used for WPA/WPA2 Personal/Enterprise security options.
static short	**COUNTRY_CODE_CUSTOM** Specifies the use of custom channels.
static short	**COUNTRY_CODE_DEFAULT** Specifies the default country code for node operation.
static short	**EFFISTREAM_MATCH_ETH_DST** Specifies a Effistream™ match code for the ETHERNET destination address field. The matchCriteria of the EffistreamRule specifies a string containing a MAC-address.
static short	**EFFISTREAM_MATCH_ETH_SRC** Specifies a Effistream™ match code for the ETHERNET source address field. The matchCriteria of the EffistreamRule specifies a string containing a MAC-address.
static short	**EFFISTREAM_MATCH_ETH_TYPE** Specifies a Effistream™ match code for the ETHERNET type field.
static short	**EFFISTREAM_MATCH_IGNORE** Specifies a Effistream™ match code used at the ROOT level. The matchCriteria of the EffistreamRule specifies a string containing an integer.
static short	**EFFISTREAM_MATCH_IP_DIFFSRV** Specifies a Effistream™ match code for the IP Diffrentiated services field. The matchCriteria of the EffistreamRule specifies a string containing an integer.
static short	**EFFISTREAM_MATCH_IP_DST**

	Specifies a Effistream™ match code for the IP destination address field. The `matchCriteria` of the `EffistreamRule` specifies a string containing a IP-address.
`static short`	**`EFFISTREAM_MATCH_IP_PROTO`** Specifies a Effistream™ match code for the IP protocol field. The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.
`static short`	**`EFFISTREAM_MATCH_IP_SRC`** Specifies a Effistream™ match code for the IP source address field. The `matchCriteria` of the `EffistreamRule` specifies a string containing a IP-address.
`static short`	**`EFFISTREAM_MATCH_IP_TOS`** Specifies a Effistream™ match code for the IP Type-of-Service field. The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.
`static short`	**`EFFISTREAM_MATCH_RTP_LENGTH`** Specifies a Effistream™ match code for the RTP data length. The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).
`static short`	**`EFFISTREAM_MATCH_RTP_PAYLOAD`** Specifies a Effistream™ match code for the RTP payload code field. The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.
`static short`	**`EFFISTREAM_MATCH_RTP_VERSION`** Specifies a Effistream™ match code for the RTP version field. The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.
`static short`	**`EFFISTREAM_MATCH_TCP_DST_PORT`** Specifies a Effistream™ match code for the TCP destination port field. The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).
`static short`	**`EFFISTREAM_MATCH_TCP_LENGTH`** Specifies a Effistream™ match code for the TCP segment length. The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).
`static short`	**`EFFISTREAM_MATCH_TCP_SRC_PORT`** Specifies a Effistream™ match code for the TCP source port field. The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).
`static short`	**`EFFISTREAM_MATCH_UDP_DST_PORT`** Specifies a Effistream™ match code for the UDP destination port field. The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).
`static short`	**`EFFISTREAM_MATCH_UDP_LENGTH`** Specifies a Effistream™ match code for the UDP datagram length. The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).
`static short`	**`EFFISTREAM_MATCH_UDP_SRC_PORT`** Specifies a Effistream™ match code for the UDP source port field. The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

static int	**EVENT_NETWORK_CLOSE** Specifies that a network was closed.
static int	**EVENT_NODE_DEAD** Specifies that a node in unreachable in the mesh network.
static int	**EVENT_NODE_HEARTBEAT** Specifies that a heartbeat was received from a node in the mesh network.
static int	**EVENT_NODE_HEARTBEAT_MISS** Specifies that a node's heartbeat was missed in the mesh network.
static int	**EVENT_NODE_SCAN** Specifies that a node is conducting dynamic channel allocation scan.
static short	**MG_CLIENT_MODE_FORWARDER** Specifies that the Meshdynamics Management Gateway client operates as a packet forwader, forwarding all management packets from the `Node`'s to the server.
static short	**MG_CLIENT_MODE_REMOTE_MANAGER** Specifies that the Meshdynamics Management Gateway client operates as a remote manager, receiving management packets from remote sites.
static short	**NETWORK_TYPE_FIPS_140_2** Specifies that the mesh network is a FIPS 140-2 secure network.
static short	**NETWORK_TYPE_REGULAR** Specifies that the mesh network is a regular network.
static short	**OPTION_ADHOC** Specifies that a `Node` has the Disjoint Adhoc feature option turned on.
static short	**OPTION_ADHOC_DHCP** Specifies that a `Node` has the DHCP server option turned on.
static short	**OPTION_ADHOC_INFRA_BEGIN** Specifies that a `Node` has the 'Begin in infrastructure' option turned on for the Disjoint Adhoc feature.
static short	**OPTION_ADHOC_SECTORED** Specifies that a `Node` has the 'SECTORED arbitration' option turned on for the Disjoint Adhoc feature.
static short	**OPTION_FORCED_ROOT** Specifies that a `Node` has the Forced Root feature option turned on.
static short	**OPTION_IGMP** Specifies that a `Node` has the IGMP multicast optimization option turned on.
static short	**OPTION_LOCATION** Specifies that a `Node` has the 802.11 PROBE request based location tracking turned on.
static short	**OPTION_SIP** Specifies that a `Node` has the 'SIP PHONE SYSTEM' option turned on.
static short	**PERFORMANCE_PROTOCOL_TCP** Specifies usage of TCP protocol for running performance tests on a `Node`.
static short	**PERFORMANCE_PROTOCOL_UDP** Specifies usage of UDP protocol for running performance tests on a `Node`.
static short	**PERFORMANCE_TYPE_DUAL_INDIVIDUAL** Specifies that performance tests on a `Node` be run in the direction `Host -> Node` and then `Node -> Host`.

static short	**PERFORMANCE_TYPE_DUAL_SIMULTANEOUS** Specifies that performance tests on a `Node` be run in the direction `Host -> Node` and `Node -> Host` simultaneously.
static short	**PERFORMANCE_TYPE_SINGLE** Specifies that performance tests on a `Node` be run in the direction `Host -> Node`.
static short	**PHY_SUB_TYPE_802_11_A** Specifies that the `InterfaceConfiguration` object contains information about a IEEE 802.11a interface.
static short	**PHY_SUB_TYPE_802_11_B** Specifies that the `InterfaceConfiguration` object contains information about a IEEE 802.11b interface.
static short	**PHY_SUB_TYPE_802_11_BG** Specifies that the `InterfaceConfiguration` object contains information about a mixed mode IEEE 802.11b/g interface.
static short	**PHY_SUB_TYPE_802_11_G** Specifies that the `InterfaceConfiguration` object contains information about a IEEE 802.11g interface.
static short	**PHY_SUB_TYPE_802_11_PSF** Specifies that the `InterfaceConfiguration` object contains information about a 20 MHz channel-width 4.9GHz interface.
static short	**PHY_SUB_TYPE_802_11_PSH** Specifies that the `InterfaceConfiguration` object contains information about a 10 MHz channel-width 4.9GHz interface.
static short	**PHY_SUB_TYPE_802_11_PSQ** Specifies that the `InterfaceConfiguration` object contains information about a 5 MHz channel-width 4.9GHz interface.
static short	**PHY_SUB_TYPE_IGNORE** Specifies that the `InterfaceConfiguration` object contains information about an ETHERNET interface. For interfaces with a `phyType` value of `PHY_TYPE_ETHERNET`, the `phySubType` shall be `PHY_SUB_TYPE_IGNORE`.
static short	**PHY_TYPE_802_11** Specifies that the `InterfaceConfiguration` object contains information about a IEEE 802.11 wireless interface.
static short	**PHY_TYPE_ETHERNET** Specifies that the `InterfaceConfiguration` object contains information about an ETHERNET interface.
static short	**REG_DOMAIN_CODE_CUSTOM** Speciies the custom regulatory domain for node operation.
static short	**REG_DOMAIN_CODE_ETSI** Specifies the ETSI regulatory domain for node operation.
static short	**REG_DOMAIN_CODE_FCC** Specifies the FCC regulatory domain for node operation.
static short	**REG_DOMAIN_CODE_NONE** Specifies a NULL regulatory domain for node operation.
static short	**SECURITY_TYPE_NONE** `InterfaceConfiguration`

	Specifies that the object contains no security parameters. With this setting the `securityInfo` field of the `InterfaceConfiguration` is ignored and set to `null`.
`static short`	**`SECURITY_TYPE_WEP_104`** Specifies that the `InterfaceConfiguration` object contains security parameters for IEEE 802.11 WEP encryption using a 104-bit key. With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WEPSecurity` object.
`static short`	**`SECURITY_TYPE_WEP_40`** Specifies that the `InterfaceConfiguration` object contains security parameters for IEEE 802.11 WEP encryption using a 40-bit key. With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WEPSecurity` object.
`static short`	**`SECURITY_TYPE_WPA_ENTERPRISE`** Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access encryption using a RADIUS server. With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAEnterpriseSecurity` object.
`static short`	**`SECURITY_TYPE_WPA_PERSONAL`** Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access encryption using a pre-shared key. With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAPersonalSecurity` object.
`static short`	**`SECURITY_TYPE_WPA2_ENTERPRISE`** Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access 2 encryption using a RADIUS server. With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAEnterpriseSecurity` object.
`static short`	**`SECURITY_TYPE_WPA2_PERSONAL`** Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access 2 encryption using a pre-shared key. With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAPersonalSecurity` object.
`static short`	**`USAGE_TYPE_DOWNLINK`** Specifies that the `InterfaceConfiguration` object contains information about a DOWNLINK interface.
`static short`	**`USAGE_TYPE_SCANNER`** Specifies that the `InterfaceConfiguration` object contains information about a SCANNER interface.
`static short`	**`USAGE_TYPE_UPLINK`** Specifies that the `InterfaceConfiguration` object contains information about an UPLINK interface.

Constructor Summary

protected	**NMS**() Protected default constructor to be used by derived classes.

Method Summary

static java.lang.String	**bytesToHexString**(byte[] bytes) This utility method converts a byte array to a hexadecimal string.
abstract int	**closeNetwork**(NMS.Network network) Closes the specified network.
static NMS	**getInstance**() Returns a reference to the singleton instance of the NMS class.
abstract NMS.Network	**getNetworkByName**(java.lang.String networkName) Returns a reference to a Network object with the specified identifier.
abstract java.util.Enumeration<NMS.Network>	**getOpenNetworks**() Returns an Enumeration of all open Network objects.
static byte[]	**hexStringToBytes**(java.lang.String hexString) This utility method converts a hexadecimal string into a byte array.
static java.lang.String	**ipAddressBytesToString**(byte[] ipAddress) This utlity method converts a byte representation of IP-address to a dotted decimal dormat string.
static byte[]	**ipAddressStringToBytes**(java.lang.String ipAddress) This utility method converts a dotted-decimal format string IP-address to an array of bytes.
static java.lang.String	**macAddressBytesToHexString**(byte[] macAddress) This utility method converts a byte representation of MAC-address to a string where the individual bytes are seperated by a ':' character.
static byte[]	**macAddressHexStringToBytes**(java.lang.String macAddress) This utility method converts a string repsentation of MAC-address to an array of bytes.
abstract NMS.Network	**openNetwork**(java.lang.String networkName, java.lang.String networkKey, int networkType) Opens the specified mesh network.
abstract int	**start**() Starts the node detection and event generation processes for the NMS object.
abstract int	**startMGClient**(short mode, java.lang.String server, int port, boolean useSSL, java.lang.String userName, java.lang.String password, boolean ignoreLocalPackets) Starts the Meshdynamics Management Gateway client for remote management.
abstract void	**stdErrPrintln**(java.lang.String str) Prints the specified string to the error output stream.
abstract void	**stdOutPrintln**(java.lang.String str) Prints the specified string to the standard output stream.
abstract int	**stop**()

	Stops the node detection and event generation processes for the NMS object.
abstract int	**stopMGClient**() Stops the Meshdynamics Management Gateway client for remote management.
protected abstract void	**unInitialize**() Un-initializes the NMS instance.
static void	**unInitializeInstance**() Un-initializes the singleton instance of the NMS class.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

USAGE_TYPE_UPLINK

```
public static final short USAGE_TYPE_UPLINK
```

Specifies that the InterfaceConfiguration object contains information about an UPLINK interface.

See Also:
NMS.InterfaceConfiguration.usageType, Constant Field Values

USAGE_TYPE_DOWNLINK

```
public static final short USAGE_TYPE_DOWNLINK
```

Specifies that the InterfaceConfiguration object contains information about a DOWNLINK interface.

See Also:
NMS.InterfaceConfiguration.usageType, Constant Field Values

USAGE_TYPE_SCANNER

```
public static final short USAGE_TYPE_SCANNER
```

Specifies that the InterfaceConfiguration object contains information about a SCANNER interface.

See Also:
NMS.InterfaceConfiguration.usageType, Constant Field Values

PHY_TYPE_ETHERNET

```
public static final short PHY_TYPE_ETHERNET
```

Specifies that the InterfaceConfiguration object contains information about an ETHERNET interface.

See Also:
NMS.InterfaceConfiguration.phyType, Constant Field Values

PHY_TYPE_802_11

```
public static final short PHY_TYPE_802_11
```

Specifies that the InterfaceConfiguration object contains information about a IEEE 802.11 wireless interface.

See Also:
NMS.InterfaceConfiguration.phyType, Constant Field Values

PHY_SUB_TYPE_IGNORE

```
public static final short PHY_SUB_TYPE_IGNORE
```

Specifies that the InterfaceConfiguration object contains information about an ETHERNET interface.
For interfaces with a phyType value of PHY_TYPE_ETHERNET, the phySubType shall be PHY_SUB_TYPE_IGNORE.

See Also:
NMS.InterfaceConfiguration.phyType, NMS.InterfaceConfiguration.phySubType, Constant Field Values

---

PHY_SUB_TYPE_802_11_A

```
public static final short PHY_SUB_TYPE_802_11_A
```

Specifies that the InterfaceConfiguration object contains information about a IEEE 802.11a interface.

See Also:
NMS.InterfaceConfiguration.phySubType, Constant Field Values

PHY_SUB_TYPE_802_11_B

```
public static final short PHY_SUB_TYPE_802_11_B
```

Specifies that the InterfaceConfiguration object contains information about a IEEE 802.11b interface.

See Also:
NMS.InterfaceConfiguration.phySubType, Constant Field Values

PHY_SUB_TYPE_802_11_G

```
public static final short PHY_SUB_TYPE_802_11_G
```

Specifies that the InterfaceConfiguration object contains information about a IEEE 802.11g interface.

See Also:
NMS.InterfaceConfiguration.phySubType, Constant Field Values

PHY_SUB_TYPE_802_11_BG

`public static final short` **`PHY_SUB_TYPE_802_11_BG`**

Specifies that the `InterfaceConfiguration` object contains information about a mixed mode IEEE 802.11b/g interface.

See Also:
`NMS.InterfaceConfiguration.phySubType`, Constant Field Values

PHY_SUB_TYPE_802_11_PSQ

`public static final short` **`PHY_SUB_TYPE_802_11_PSQ`**

Specifies that the `InterfaceConfiguration` object contains information about a 5 MHz channel-width 4.9GHz interface.

See Also:
`NMS.InterfaceConfiguration.phySubType`, Constant Field Values

PHY_SUB_TYPE_802_11_PSH

`public static final short` **`PHY_SUB_TYPE_802_11_PSH`**

Specifies that the `InterfaceConfiguration` object contains information about a 10 MHz channel-width 4.9GHz interface.

See Also:
`NMS.InterfaceConfiguration.phySubType`, Constant Field Values

PHY_SUB_TYPE_802_11_PSF

`public static final short` **`PHY_SUB_TYPE_802_11_PSF`**

Specifies that the `InterfaceConfiguration` object contains information about a 20 MHz channel-width 4.9GHz interface.

See Also:
`NMS.InterfaceConfiguration.phySubType`, Constant Field Values

SECURITY_TYPE_NONE

`public static final short` **`SECURITY_TYPE_NONE`**

Specifies that the `InterfaceConfiguration` object contains no security parameters.

With this setting the `securityInfo` field of the `InterfaceConfiguration` is ignored and set to `null`.

See Also:
`NMS.InterfaceConfiguration.securityType`, Constant Field Values

SECURITY_TYPE_WEP_40

`public static final short` **`SECURITY_TYPE_WEP_40`**

Specifies that the `InterfaceConfiguration` object contains security parameters for IEEE 802.11 WEP encryption using a 40-bit key.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WEPSecurity` object.

See Also:
`NMS.InterfaceConfiguration.securityType`, `NMS.WEPSecurity`, Constant Field Values

SECURITY_TYPE_WEP_104

`public static final short` **`SECURITY_TYPE_WEP_104`**

Specifies that the `InterfaceConfiguration` object contains security parameters for IEEE 802.11 WEP encryption using a 104-bit key.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WEPSecurity` object.

See Also:
`NMS.InterfaceConfiguration.securityType`, `NMS.WEPSecurity`, Constant Field Values

SECURITY_TYPE_WPA_PERSONAL

`public static final short` **`SECURITY_TYPE_WPA_PERSONAL`**

Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access encryption using a pre-shared key.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAPersonalSecurity` object.

See Also:
`NMS.InterfaceConfiguration.securityType`, `NMS.WPAPersonalSecurity`, Constant Field Values

SECURITY_TYPE_WPA_ENTERPRISE

`public static final short` **`SECURITY_TYPE_WPA_ENTERPRISE`**

Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access encryption using a RADIUS server.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAEnterpriseSecurity` object.

See Also:
`NMS.InterfaceConfiguration.securityType`, `NMS.WPAEnterpriseSecurity`, Constant Field Values

SECURITY_TYPE_WPA2_PERSONAL

`public static final short` **`SECURITY_TYPE_WPA2_PERSONAL`**

Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access 2 encryption using a pre-shared key.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAPersonalSecurity` object.

See Also:
`NMS.InterfaceConfiguration.securityType`, `NMS.WPAPersonalSecurity`, Constant Field Values

SECURITY_TYPE_WPA2_ENTERPRISE

`public static final short` **`SECURITY_TYPE_WPA2_ENTERPRISE`**

Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access 2 encryption using a RADIUS server.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAEnterpriseSecurity` object.

See Also:
`NMS.InterfaceConfiguration.securityType`, `NMS.WPAEnterpriseSecurity`, Constant Field Values

CIPHER_CCMP

`public static final short` **`CIPHER_CCMP`**

Specifies AES-CCMP encryption to be used for WPA/WPA2 Personal/Enterprise security options.

See Also:
`NMS.InterfaceConfiguration.securityType`, `NMS.WPAPersonalSecurity.cipherType`, `NMS.WPAEnterpriseSecurity.cipherType`, Constant Field Values

CIPHER_TKIP

`public static final short` **`CIPHER_TKIP`**

Specifies TKIP encryption to be used for WPA/WPA2 Personal/Enterprise security options.

See Also:
`NMS.InterfaceConfiguration.securityType`, `NMS.WPAPersonalSecurity.cipherType`, `NMS.WPAEnterpriseSecurity.cipherType`, Constant Field Values

EVENT_NODE_HEARTBEAT

`public static final int` **`EVENT_NODE_HEARTBEAT`**

Specifies that a heartbeat was received from a node in the mesh network.

See Also:
NMS.NetworkListener.onEvent(int, com.meshdynamics.api.NMS.Network, com.meshdynamics.api.NMS.Node), Constant Field Values

---

EVENT_NODE_HEARTBEAT_MISS

```
public static final int EVENT_NODE_HEARTBEAT_MISS
```

Specifies that a node's heartbeat was missed in the mesh network.

See Also:
NMS.NetworkListener.onEvent(int, com.meshdynamics.api.NMS.Network, com.meshdynamics.api.NMS.Node), Constant Field Values

---

EVENT_NODE_DEAD

```
public static final int EVENT_NODE_DEAD
```

Specifies that a node in unreachable in the mesh network.

See Also:
NMS.NetworkListener.onEvent(int, com.meshdynamics.api.NMS.Network, com.meshdynamics.api.NMS.Node), Constant Field Values

---

EVENT_NODE_SCAN

```
public static final int EVENT_NODE_SCAN
```

Specifies that a node is conducting dynamic channel allocation scan.

See Also:
NMS.NetworkListener.onEvent(int, com.meshdynamics.api.NMS.Network, com.meshdynamics.api.NMS.Node), Constant Field Values

---

EVENT_NETWORK_CLOSE

```
public static final int EVENT_NETWORK_CLOSE
```

Specifies that a network was closed.

See Also:
NMS.NetworkListener.onEvent(int, com.meshdynamics.api.NMS.Network, com.meshdynamics.api.NMS.Node), Constant Field Values

---

OPTION_IGMP

```
public static final short OPTION_IGMP
```

Specifies that a `Node` has the IGMP multicast optimization option turned on.

See Also:
NMS.GeneralConfiguration.options, Constant Field Values

OPTION_ADHOC

```
public static final short OPTION_ADHOC
```

Specifies that a Node has the Disjoint Adhoc feature option turned on.

See Also:
NMS.GeneralConfiguration.options, Constant Field Values

OPTION_FORCED_ROOT

```
public static final short OPTION_FORCED_ROOT
```

Specifies that a Node has the Forced Root feature option turned on.

See Also:
NMS.GeneralConfiguration.options, Constant Field Values

OPTION_ADHOC_INFRA_BEGIN

```
public static final short OPTION_ADHOC_INFRA_BEGIN
```

Specifies that a Node has the 'Begin in infrastructure' option turned on for the Disjoint Adhoc feature.

See Also:
NMS.GeneralConfiguration.options, Constant Field Values

OPTION_ADHOC_DHCP

```
public static final short OPTION_ADHOC_DHCP
```

Specifies that a Node has the DHCP server option turned on.

See Also:
NMS.GeneralConfiguration.options, Constant Field Values

OPTION_LOCATION

```
public static final short OPTION_LOCATION
```

Specifies that a Node has the 802.11 PROBE request based location tracking turned on.

See Also:
NMS.GeneralConfiguration.options, Constant Field Values

OPTION_ADHOC_SECTORED

`public static final short` **`OPTION_ADHOC_SECTORED`**

Specifies that a `Node` has the 'SECTORED arbitration' option turned on for the Disjoint Adhoc feature.

See Also:
`NMS.GeneralConfiguration.options`, Constant Field Values

OPTION_SIP

`public static final short` **`OPTION_SIP`**

Specifies that a `Node` has the 'SIP PHONE SYSTEM' option turned on.

See Also:
`NMS.GeneralConfiguration.options`, Constant Field Values

NETWORK_TYPE_REGULAR

`public static final short` **`NETWORK_TYPE_REGULAR`**

Specifies that the mesh network is a regular network.

See Also:
`openNetwork(java.lang.String, java.lang.String, int)`, Constant Field Values

NETWORK_TYPE_FIPS_140_2

`public static final short` **`NETWORK_TYPE_FIPS_140_2`**

Specifies that the mesh network is a FIPS 140-2 secure network.

See Also:
`openNetwork(java.lang.String, java.lang.String, int)`, Constant Field Values

EFFISTREAM_MATCH_IGNORE

`public static final short` **`EFFISTREAM_MATCH_IGNORE`**

Specifies a Effistream™ match code used at the ROOT level.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_ETH_TYPE

`public static final short` **`EFFISTREAM_MATCH_ETH_TYPE`**

Specifies a Effistream^{TM} match code for the ETHERNET type field.

See Also:
NMS.EffistreamRule.matchId, NMS.EffistreamRule.matchCriteria, Constant Field Values

---

EFFISTREAM_MATCH_ETH_DST

```
public static final short EFFISTREAM_MATCH_ETH_DST
```

Specifies a Effistream^{TM} match code for the ETHERNET destination address field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a MAC-address.

See Also:
NMS.EffistreamRule.matchId, NMS.EffistreamRule.matchCriteria, Constant Field Values

---

EFFISTREAM_MATCH_ETH_SRC

```
public static final short EFFISTREAM_MATCH_ETH_SRC
```

Specifies a Effistream^{TM} match code for the ETHERNET source address field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a MAC-address.

See Also:
NMS.EffistreamRule.matchId, NMS.EffistreamRule.matchCriteria, Constant Field Values

---

EFFISTREAM_MATCH_IP_TOS

```
public static final short EFFISTREAM_MATCH_IP_TOS
```

Specifies a Effistream^{TM} match code for the IP Type-of-Service field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.

See Also:
NMS.EffistreamRule.matchId, NMS.EffistreamRule.matchCriteria, Constant Field Values

---

EFFISTREAM_MATCH_IP_DIFFSRV

```
public static final short EFFISTREAM_MATCH_IP_DIFFSRV
```

Specifies a Effistream^{TM} match code for the IP Diffrentiated services field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.

See Also:
NMS.EffistreamRule.matchId, NMS.EffistreamRule.matchCriteria, Constant Field Values

---

EFFISTREAM_MATCH_IP_SRC

```
public static final short EFFISTREAM_MATCH_IP_SRC
```

Specifies a Effistream™ match code for the IP source address field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a IP-address.

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_IP_DST

```
public static final short EFFISTREAM_MATCH_IP_DST
```

Specifies a Effistream™ match code for the IP destination address field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a IP-address.

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_IP_PROTO

```
public static final short EFFISTREAM_MATCH_IP_PROTO
```

Specifies a Effistream™ match code for the IP protocol field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_UDP_SRC_PORT

```
public static final short EFFISTREAM_MATCH_UDP_SRC_PORT
```

Specifies a Effistream™ match code for the UDP source port field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_UDP_DST_PORT

```
public static final short EFFISTREAM_MATCH_UDP_DST_PORT
```

Specifies a Effistream™ match code for the UDP destination port field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_UDP_LENGTH

```
public static final short
```

EFFISTREAM_MATCH_UDP_LENGTH

Specifies a Effistream™ match code for the UDP datagram length.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_TCP_SRC_PORT

`public static final short` **`EFFISTREAM_MATCH_TCP_SRC_PORT`**

Specifies a Effistream™ match code for the TCP source port field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_TCP_DST_PORT

`public static final short` **`EFFISTREAM_MATCH_TCP_DST_PORT`**

Specifies a Effistream™ match code for the TCP destination port field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_TCP_LENGTH

`public static final short` **`EFFISTREAM_MATCH_TCP_LENGTH`**

Specifies a Effistream™ match code for the TCP segment length.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_RTP_VERSION

`public static final short` **`EFFISTREAM_MATCH_RTP_VERSION`**

Specifies a Effistream™ match code for the RTP version field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_RTP_PAYLOAD

`public static final short` **`EFFISTREAM_MATCH_RTP_PAYLOAD`**

Specifies a Effistream™ match code for the RTP payload code field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_RTP_LENGTH

`public static final short` **`EFFISTREAM_MATCH_RTP_LENGTH`**

Specifies a Effistream™ match code for the RTP data length.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

PERFORMANCE_PROTOCOL_TCP

`public static final short` **`PERFORMANCE_PROTOCOL_TCP`**

Specifies usage of TCP protocol for running performance tests on a `Node`.

See Also:
`NMS.Node.runPerformanceTest(int, short, short, int)`, Constant Field Values

PERFORMANCE_PROTOCOL_UDP

`public static final short` **`PERFORMANCE_PROTOCOL_UDP`**

Specifies usage of UDP protocol for running performance tests on a `Node`.

See Also:
`NMS.Node.runPerformanceTest(int, short, short, int)`, Constant Field Values

PERFORMANCE_TYPE_SINGLE

`public static final short` **`PERFORMANCE_TYPE_SINGLE`**

Specifies that performance tests on a `Node` be run in the direction `Host -> Node`.

See Also:
`NMS.Node.runPerformanceTest(int, short, short, int)`, Constant Field Values

PERFORMANCE_TYPE_DUAL_INDIVIDUAL

`public static final short` **`PERFORMANCE_TYPE_DUAL_INDIVIDUAL`**

Specifies that performance tests on a `Node` be run in the direction `Host -> Node` and then `Node -> Host`.

See Also:
`NMS.Node.runPerformanceTest(int, short, short, int)`, Constant Field Values

PERFORMANCE_TYPE_DUAL_SIMULTANEOUS

`public static final short` **`PERFORMANCE_TYPE_DUAL_SIMULTANEOUS`**

Specifies that performance tests on a `Node` be run in the direction `Host -> Node` and `Node -> Host` simultaneously.

See Also:
`NMS.Node.runPerformanceTest(int, short, short, int)`, Constant Field Values

MG_CLIENT_MODE_FORWARDER

`public static final short` **`MG_CLIENT_MODE_FORWARDER`**

Specifies that the Meshdynamics Management Gateway client operates as a packet forwader, forwarding all management packets from the `Node`'s to the server.

See Also:
`startMGClient(short, java.lang.String, int, boolean, java.lang.String, java.lang.String, boolean)`, Constant Field Values

MG_CLIENT_MODE_REMOTE_MANAGER

`public static final short` **`MG_CLIENT_MODE_REMOTE_MANAGER`**

Specifies that the Meshdynamics Management Gateway client operates as a remote manager, receiving management packets from remote sites.

See Also:
`startMGClient(short, java.lang.String, int, boolean, java.lang.String, java.lang.String, boolean)`, Constant Field Values

COUNTRY_CODE_DEFAULT

`public static final short` **`COUNTRY_CODE_DEFAULT`**

Specifies the default country code for node operation.

See Also:
Constant Field Values

COUNTRY_CODE_CUSTOM

`public static final short` **`COUNTRY_CODE_CUSTOM`**

Specifies the use of custom channels.

This is only allowed via the use of the Meshdynamics RF-Editor API.

See Also:
Constant Field Values

REG_DOMAIN_CODE_NONE

```
public static final short REG_DOMAIN_CODE_NONE
```

Specifies a NULL regulatory domain for node operation.

See Also:
Constant Field Values

REG_DOMAIN_CODE_FCC

```
public static final short REG_DOMAIN_CODE_FCC
```

Specifies the FCC regulatory domain for node operation.

See Also:
Constant Field Values

REG_DOMAIN_CODE_ETSI

```
public static final short REG_DOMAIN_CODE_ETSI
```

Specifies the ETSI regulatory domain for node operation.

See Also:
Constant Field Values

REG_DOMAIN_CODE_CUSTOM

```
public static final short REG_DOMAIN_CODE_CUSTOM
```

Speciies the custom regulatory domain for node operation.

This is only allowed via the use of the Meshdynamics RF-Editor API.

See Also:
Constant Field Values

Constructor Detail

NMS

```
protected NMS()
```

Protected default constructor to be used by derived classes.

Method Detail

getInstance

```
public static NMS getInstance()
```

Returns a reference to the singleton instance of the `NMS` class.

Returns:
reference to the `NMS` instance

---

unInitializeInstance

```
public static void unInitializeInstance()
```

Un-initializes the singleton instance of the `NMS` class.

---

hexStringToBytes

```
public static byte[] hexStringToBytes(java.lang.String hexString)
```

This utility method converts a hexadecimal string into a byte array.

Parameters:
`hexString` - the hexadecimal string
Returns:
byte array containing the byte representation of the hexadecimal string
See Also:
`bytesToHexString(byte[])`

---

bytesToHexString

```
public static java.lang.String bytesToHexString(byte[] bytes)
```

This utility method converts a byte array to a hexadecimal string.

Parameters:
`bytes` - the byte array to be converted.
Returns:
hexadecimal string
See Also:
`hexStringToBytes(java.lang.String)`

---

macAddressBytesToHexString

```
public static java.lang.String macAddressBytesToHexString(byte[] macAddress)
```

This utility method converts a byte representation of MAC-address to a string where the individual bytes are

seperated by a ':' character.

Parameters:
`macAddress` - byte array containing the MAC address
Returns:
string representation of the MAC address
See Also:
`macAddressHexStringToBytes(java.lang.String)`

---

ipAddressBytesToString

```
public static java.lang.String ipAddressBytesToString(byte[] ipAddress)
```

This utlity method converts a byte representation of IP-address to a dotted decimal dormat string.

Parameters:
`ipAddress` - the byte array containing the IP-address
Returns:
dotted decimal format string representation of the IP-address
See Also:
`ipAddressStringToBytes(java.lang.String)`

---

macAddressHexStringToBytes

```
public static byte[] macAddressHexStringToBytes(java.lang.String macAddress)
```

This utility method converts a string repsentation of MAC-address to an array of bytes.

Parameters:
`macAddress` - the string representation of the MAC-address.
Returns:
byte array containing the MAC-address
See Also:
`macAddressBytesToHexString(byte[])`

---

ipAddressStringToBytes

```
public static byte[] ipAddressStringToBytes(java.lang.String ipAddress)
```

This utility method converts a dotted-decimal format string IP-address to an array of bytes.

Parameters:
`ipAddress` - the dotted-decimal string IP-address.
Returns:
byte array containing the IP-address
See Also:
`ipAddressBytesToString(byte[])`

---

start

```
public abstract int start()
```

Starts the node detection and event generation processes for the `NMS` object.

Returns:

`0` on success

---

stop

```
public abstract int stop()
```

Stops the node detection and event generation processes for the `NMS` object.

Returns:

`0` on success

---

startMGClient

```
public abstract int startMGClient(short mode,
                                  java.lang.String server,
                                  int port,
                                  boolean useSSL,
                                  java.lang.String userName,
                                  java.lang.String password,
                                  boolean ignoreLocalPackets)
```

Starts the Meshdynamics Management Gateway client for remote management.

The Meshdynamics Management Gateway client connects to a Meshdynamics Management Gateway server using the HTTP protocol.

Parameters:

`mode` - the client mode, can be one of `MG_CLIENT_MODE_FORWARDER` or `MG_CLIENT_MODE_REMOTE_MANAGER`
`server` - the IP address or host name of the Meshdynamics Management Gateway server
`port` - the port on which the Meshdynamics Management Gateway server listens
`useSSL` - set to `true` if a SSL connection is to be used
`userName` - the account user-name at the Meshdynamics Management Gateway server
`password` - the account password
`ignoreLocalPackets` - local incoming packets will be ignored in `MG_CLIENT_MODE_REMOTE_MANAGER` mode

Returns:

`0` on success

---

stopMGClient

```
public abstract int stopMGClient()
```

Stops the Meshdynamics Management Gateway client for remote management.

Returns:

`0` on success

---

openNetwork

```
public abstract NMS.Network            (java.lang.String networkName,
```

```
openNetwork
            java.lang.String networkKey,
            int networkType)
```

Opens the specified mesh network.

Parameters:

`networkName` - the mesh network identifier
`networkKey` - the mesh network key
`networkType` - the network type (`NMS.NETWORK_TYPE_REGULAR` or `NMS.NETWORK_TYPE_FIPS_140_2`). For `NMS.NETWORK_TYPE_FIPS_140_2` the `networkKey` specifies a 128-bit hexstring.

Returns:

reference to the `Network` object or `null` on failure

---

closeNetwork

```
public abstract int closeNetwork(NMS.Network network)
```

Closes the specified network.

Parameters:

`network` - the mesh network to be closed

Returns:

`0` on success

---

getOpenNetworks

```
public abstract java.util.Enumeration<NMS.Network> getOpenNetworks()
```

Returns an `Enumeration` of all open `Network` objects.

Returns:

`Enumeration` of all open `Network` objects.

---

getNetworkByName

```
public abstract NMS.Network getNetworkByName(java.lang.String networkName)
```

Returns a reference to a `Network` object with the specified identifier.

Parameters:

`networkName` - the mesh network identifier

Returns:

reference to the `Network` object or `null` on failure

---

stdOutPrintln

```
public abstract void stdOutPrintln(java.lang.String str)
```

Prints the specified string to the standard output stream.

Parameters:

`str` - the string to be printed

---

stdErrPrintln

```
public abstract void stdErrPrintln(java.lang.String str)
```

Prints the specified string to the error output stream.

Parameters:

`str` - the string to be printed

---

unInitialize

```
protected abstract void unInitialize()
```

Un-initializes the `NMS` instance.

---

com.meshdynamics.api

Class NMS.ACLConfiguration

```
java.lang.Object
  └ com.meshdynamics.api.NMS.ACLConfiguration
```

Enclosing class:
NMS

```
public static class NMS.ACLConfiguration
extends java.lang.Object
```

Defines the Access Control List configuration for a node.

Field Summary

NMS.ObjectArray	**entries** The array of NMS.ACLEntry objects.
short	**whiteList** Defines whether the ACL configuration entries specify a 'white-list'.

Constructor Summary

NMS.ACLConfiguration() Default constructor, initializes the object with an empty entries array and sets whiteList to 0.
NMS.ACLConfiguration(java.lang.String objectNotation) Constructs the ACLConfiguration from a object notation string.

Method Summary

void	**addEntry**(NMS.ACLEntry entry) Adds the entry into the entries array.
java.lang.String	**toObjectNotation**() Returns a string containing the object notation representation of the ACLConfiguration object.
java.lang.String	**toString**()

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

whiteList

`public short` **`whiteList`**

Defines whether the ACL configuration entries specify a 'white-list'.

If non-zero, the entries are used as a white-list i.e clients that are not in the list shall be rejected.

---

entries

`public` `NMS.ObjectArray` **`entries`**

The array of `NMS.ACLEntry` objects.

Constructor Detail

NMS.ACLConfiguration

`public` **`NMS.ACLConfiguration`**`()`

Default constructor, initializes the object with an empty entries array and sets `whiteList` to `0`.

NMS.ACLConfiguration

`public` **`NMS.ACLConfiguration`**`(java.lang.String objectNotation)`

Constructs the `ACLConfiguration` from a object notation string.

Parameters:

`objectNotation` -

Method Detail

toString

`public java.lang.String` **`toString`**`()`

Overrides:

`toString` in class `java.lang.Object`

---

toObjectNotation

`public java.lang.String` **`toObjectNotation`**`()`

Returns a string containing the object notation representation of the ACLConfiguration object.

Returns:

the object notation string

---

addEntry

```
public void addEntry(NMS.ACLEntry entry)
```

Adds the entry into the entries array.

Parameters:

`entry` - the entry to be added

---

com.meshdynamics.api

Class NMS.ACLEntry

```
java.lang.Object
  └ com.meshdynamics.api.NMS.ACLEntry
```

Enclosing class:
NMS

```
public static class NMS.ACLEntry
extends java.lang.Object
```

Defines an Access Control List entry.

Field Summary

short	**block** Set to non-zero to block the device.
short	**dot11eCategory** The IEEE 802.11e access category for the device.
short	**dot11eEnabled** Set to non-zero of `dot11eCategory` is valid.
static short	**INVALID_VLAN** Constant specifying the default VLAN.
java.lang.String	**macAddress** The MAC-address of the device.
short	**vlanTag** The IEEE 802.1q VLAN tag to be used when the device associates.

Constructor Summary

NMS.ACLEntry() Default constructor.

Method Summary

java.lang.String	**toObjectNotation**() Returns a string containing the object notation representation of the `ACLEntry` object.
java.lang.String	**toString**()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

macAddress

public java.lang.String **macAddress**

The MAC-address of the device.

---

vlanTag

public short **vlanTag**

The IEEE 802.1q VLAN tag to be used when the device associates.

Setting this value to ACLEntry.INVALID_VLAN will put the device on the default VLAN.

---

dot11eEnabled

public short **dot11eEnabled**

Set to non-zero of dot11eCategory is valid.

---

dot11eCategory

public short **dot11eCategory**

The IEEE 802.11e access category for the device.

NOTE: This field is ignored if dot11eEnabled is 0.

---

block

public short **block**

Set to non-zero to block the device.

INVALID_VLAN

public static final short **INVALID_VLAN**

Constant specifying the default VLAN.

See Also:

Constant Field Values

Constructor Detail

NMS.ACLEntry

`public` **`NMS.ACLEntry`**`()`

Default constructor.

Method Detail

toString

`public java.lang.String` **`toString`**`()`

Overrides:

`toString` in class `java.lang.Object`

---

toObjectNotation

`public java.lang.String` **`toObjectNotation`**`()`

Returns a string containing the object notation representation of the `ACLEntry` object.

Returns:

the object notation string

---

com.meshdynamics.api

Interface NMS.ConnectedDevice

Enclosing class:

NMS

`public static interface NMS.ConnectedDevice`

Defines the properties of all devices connected to a `NMS.Node`

See Also:

`NMS.Node.getConnectedDevices()`

Method Summary

`java.lang.String`	**`getMacAddress`**`()` Returns the MAC address of the device formatted as a string.
`int`	**`getRxSignal`**`()` Returns the RSSI of the packets from the device to the node.
`int`	**`getTxBitRate`**`()` Returns the transmit rate of packets from the node to the device.

Method Detail

getMacAddress

`java.lang.String` **`getMacAddress`**`()`

Returns the MAC address of the device formatted as a string.

Returns:

MAC address

---

getRxSignal

`int` **`getRxSignal`**`()`

Returns the RSSI of the packets from the device to the node.

Returns:

signal RSSI

---

getTxBitRate

`int` **`getTxBitRate`**`()`

Returns the transmit rate of packets from the node to the device.

Returns:

transmit rate

Package **Class** **Tree** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

---

com.meshdynamics.api

Class NMS.EffistreamRule

```
java.lang.Object
  └ com.meshdynamics.api.NMS.EffistreamRule
```

Enclosing class:
NMS

```
public static class NMS.EffistreamRule
extends java.lang.Object
```

Defines a Effistream QoS rule.

Field Summary

short	**actionBitRate** Specifies that the transmit rate. This field is only valid for leaf-level rules.
short	**actionDot11eCategory** Specifies that the IEEE 802.11e category.
short	**actionDropPacket** Specifies that the packets will be dropped.
short	**actionNoAck** When non-zero specifies that the packets will be sent without acknowledgement.
short	**actionQueuedRetry** Specifies that the transmit rate.
NMS.EffistreamRule	**firstChild** Reference to the next child rule object.
java.lang.String	**matchCriteria** Specifies the match criteria for the rule.
short	**matchId** Specifies the match identifier for the rule.
NMS.EffistreamRule	**nextSibling** Reference to the next sibling rule object.
NMS.EffistreamRule	**parent** Reference to the parent rule object.

Constructor Summary

NMS.EffistreamRule()
Default constructor typically used to create the 'ROOT' object for the rules.

NMS.EffistreamRule(short matchId, java.lang.String matchCriteria) Use this constructor to create a rule without specifying child rules.
NMS.EffistreamRule(short matchId, java.lang.String matchCriteria, NMS.EffistreamRule child) Use this constructor to create a rule directly specifying the first child.
NMS.EffistreamRule(short matchId, java.lang.String matchCriteria, short actionNoAck, short actionDropPacket, short actionDot11eCategory, short actionBitRate, short actionQueuedRetry) Use this constructor to create a leaf-level rule object.

Method Summary

void	**addChild**(NMS.EffistreamRule child) Adds a child rule to the rule object.
static NMS.EffistreamRule	**fromXmlSpec**(java.lang.String xmlSpec) Returns a EffistreamRule object hierarchy based on a XML based input.
java.lang.String	**toString**()
java.lang.String	**toXmlSpec**() Converts a EffistreamRule object hierarchy to a XML based string.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

matchId

public short **matchId**

Specifies the match identifier for the rule.

This can be one of NMS.EFFISTREAM_MATCH_ETH_DST,NMS.EFFISTREAM_MATCH_ETH_SRC, NMS.EFFISTREAM_MATCH_ETH_TYPE,NMS.EFFISTREAM_MATCH_IGNORE, NMS.EFFISTREAM_MATCH_IP_DIFFSRV,NMS.EFFISTREAM_MATCH_IP_DST, NMS.EFFISTREAM_MATCH_IP_PROTO,NMS.EFFISTREAM_MATCH_IP_SRC, NMS.EFFISTREAM_MATCH_IP_TOS,NMS.EFFISTREAM_MATCH_RTP_LENGTH, NMS.EFFISTREAM_MATCH_RTP_VERSION,NMS.EFFISTREAM_MATCH_TCP_DST_PORT, NMS.EFFISTREAM_MATCH_TCP_LENGTH,NMS.EFFISTREAM_MATCH_TCP_SRC_PORT, NMS.EFFISTREAM_MATCH_UDP_DST_PORT,NMS.EFFISTREAM_MATCH_UDP_LENGTH, NMS.EFFISTREAM_MATCH_UDP_SRC_PORT.

---

matchCriteria

public java.lang.String **matchCriteria**

Specifies the match criteria for the rule.

matchId

Depending on the value of this field contains either a MAC address, an IP address, a 32-bit integer or a range of 32-bit integers all formatted as a string.

For more information on the format refer to the match identifiers :

NMS.EFFISTREAM_MATCH_ETH_DST,NMS.EFFISTREAM_MATCH_ETH_SRC,
NMS.EFFISTREAM_MATCH_ETH_TYPE,NMS.EFFISTREAM_MATCH_IGNORE,
NMS.EFFISTREAM_MATCH_IP_DIFFSRV,NMS.EFFISTREAM_MATCH_IP_DST,
NMS.EFFISTREAM_MATCH_IP_PROTO,NMS.EFFISTREAM_MATCH_IP_SRC,
NMS.EFFISTREAM_MATCH_IP_TOS,NMS.EFFISTREAM_MATCH_RTP_LENGTH,
NMS.EFFISTREAM_MATCH_RTP_VERSION,NMS.EFFISTREAM_MATCH_TCP_DST_PORT,
NMS.EFFISTREAM_MATCH_TCP_LENGTH,NMS.EFFISTREAM_MATCH_TCP_SRC_PORT,
NMS.EFFISTREAM_MATCH_UDP_DST_PORT,NMS.EFFISTREAM_MATCH_UDP_LENGTH,
NMS.EFFISTREAM_MATCH_UDP_SRC_PORT

---

actionNoAck

```
public short actionNoAck
```

When non-zero specifies that the packets will be sent without acknowledgement.

This field is only valid for leaf-level rules.

---

actionDropPacket

```
public short actionDropPacket
```

Specifies that the packets will be dropped.

This field is only valid for leaf-level rules.

---

actionDot11eCategory

```
public short actionDot11eCategory
```

Specifies that the IEEE 802.11e category.

This field is only valid for leaf-level rules.

---

actionBitRate

```
public short actionBitRate
```

Specifies that the transmit rate.
This field is only valid for leaf-level rules.

---

actionQueuedRetry

```
public short actionQueuedRetry
```

Specifies that the transmit rate.

This field is only valid for leaf-level rules.

---

parent

```
public NMS.EffistreamRule parent
```

Reference to the parent rule object.

---

nextSibling

```
public NMS.EffistreamRule nextSibling
```

Reference to the next sibling rule object.

---

firstChild

```
public NMS.EffistreamRule firstChild
```

Reference to the next child rule object.

When `null`, the rule is a leaf-level rule.

Constructor Detail

NMS.EffistreamRule

```
public NMS.EffistreamRule()
```

Default constructor typically used to create the 'ROOT' object for the rules.

NMS.EffistreamRule

```
public NMS.EffistreamRule(short matchId,
                          java.lang.String matchCriteria)
```

Use this constructor to create a rule without specifying child rules.

Parameters:

`matchId` - the match identifier for the rule see `matchId`
`matchCriteria` - the criteria for a match see `matchCriteria`

NMS.EffistreamRule

```
public NMS.EffistreamRule(short matchId,
                          java.lang.String matchCriteria,
                          NMS.EffistreamRule child)
```

Use this constructor to create a rule directly specifying the first child.

e.g. `rule = new EffistreamRule(NMS.EFFISTREAM_MATCH_ETH_TYPE,"2048",new EffistreamRule(NMS.EFFISTREAM_MATCH_IP_SRC,"192.168.45.6",0,0,3,36,0)))`

Parameters:

`matchId` - the match identifier for the rule see `matchId`
`matchCriteria` - the criteria for a match see `matchCriteria`
`child` - the first child rule `firstChild`

NMS.EffistreamRule

```
public NMS.EffistreamRule(short matchId,
                          java.lang.String matchCriteria,
                          short actionNoAck,
                          short actionDropPacket,
                          short actionDot11eCategory,
                          short actionBitRate,
                          short actionQueuedRetry)
```

Use this constructor to create a leaf-level rule object.

Parameters:

`matchId` - the match identifier for the rule see `matchId`
`matchCriteria` - the criteria for a match see `matchCriteria`
`actionNoAck` - see `actionNoAck`
`actionDropPacket` - see `actionDropPacket`
`actionDot11eCategory` - see `actionDot11eCategory`
`actionBitRate` - see `actionBitRate`
`actionQueuedRetry` - see `actionQueuedRetry`

Method Detail

addChild

```
public void addChild(NMS.EffistreamRule child)
```

Adds a child rule to the rule object.

The child rule is added to the tail of the siblings list

Parameters:

`child` - the child rule to add

---

toXmlSpec

```
public java.lang.String toXmlSpec()
```

Converts a EffistreamRule object hierarchy to a XML based string.

Returns:

xml based effistream rule hierarchy

---

fromXmlSpec

```
public static NMS.EffistreamRule fromXmlSpec(java.lang.String xmlSpec)
```

Returns a EffistreamRule object hierarchy based on a XML based input.

Parameters:
`xmlSpec` - the XML input string
Returns:
a EffistreamRule object hierarchy
Throws:
`java.lang.Exception`

---

toString

```
public java.lang.String toString()
```

Overrides:
`toString` in class `java.lang.Object`

---

com.meshdynamics.api

Class NMS.GeneralConfiguration

```
java.lang.Object
  └ com.meshdynamics.api.NMS.GeneralConfiguration
```

Enclosing class:
NMS

```
public static class NMS.GeneralConfiguration
extends java.lang.Object
```

Defines all Node level fields used by a `NMS.Node`.

See Also:
`NMS.Node.getGeneralConfiguration()`,
`NMS.Node.setGeneralConfiguration(com.meshdynamics.api.NMS.GeneralConfiguration)`

Field Summary

int	**countryCode** The operating country code for the node.
short	**dfsRequired** Specifies whether Dynamics Frequency Selection and RADAR detection is required for the regulatoryDomain.
short	**dynamicChannelAllocation** The node level flag determining whether downlink interfaces shall use the Dynamic Channel Allocation scheme.
java.lang.String	**gatewayIpAddress** The ip-address of the default gateway in dotted decimal form.
java.lang.String	**gpsLatitude** Latitude coordinate of the node in decimal format.
java.lang.String	**gpsLongitude** Longitude coordinate of the node in decimal format.
short	**heartbeatInterval** The heartbeat interval for the node.
java.lang.String	**hostName** The network host-name for the node.
java.lang.String	**ipAddress** The ip-address for the node in dotted decimal form.
short	**mobilityMode** The node's mobilty mode.

java.lang.String	**model** The model identifier for the node.
java.lang.String	**nodeDescription** User-defined description for the node
java.lang.String	**nodeName** User-defined name of the node
short	**options** The combination of run-time options enabled on the node.
java.lang.String	**preferredParent** The MAC address of the preferred parent's downlink radio.
int	**regulatoryDomain** The operating regulatory domain for the node.
java.lang.String	**subnetMask** The subnet-mask for the node in dotted decimal form.

Constructor Summary

NMS.GeneralConfiguration()

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

nodeName

public java.lang.String **nodeName**

User-defined name of the node

---

nodeDescription

public java.lang.String **nodeDescription**

User-defined description for the node

---

model

public java.lang.String **model**

The model identifier for the node.

NOTE: This field is read-only and will be ignored in calls to NMS.Node.setGeneralConfiguration(com.meshdynamics.api.NMS.GeneralConfiguration).

---

gpsLatitude

`public java.lang.String` **`gpsLatitude`**

Latitude coordinate of the node in decimal format.

Coordinates South of the equator are represented by a negative number

---

gpsLongitude

`public java.lang.String` **`gpsLongitude`**

Longitude coordinate of the node in decimal format.

Coordinates West of the meridian are represented by a negative number

---

hostName

`public java.lang.String` **`hostName`**

The network host-name for the node.

---

ipAddress

`public java.lang.String` **`ipAddress`**

The ip-address for the node in dotted decimal form.

---

subnetMask

`public java.lang.String` **`subnetMask`**

The subnet-mask for the node in dotted decimal form.

---

gatewayIpAddress

`public java.lang.String` **`gatewayIpAddress`**

The ip-address of the default gateway in dotted decimal form.

---

preferredParent

`public java.lang.String` **`preferredParent`**

The MAC address of the preferred parent's downlink radio.

---

heartbeatInterval

`public short` **`heartbeatInterval`**

The heartbeat interval for the node.

---

mobilityMode

`public short` **`mobilityMode`**

The node's mobilty mode.

A non-zero value indicates that the node is configured for mobility.

---

options

`public short` **`options`**

The combination of run-time options enabled on the node.

See Also:

NMS.OPTION_ADHOC, NMS.OPTION_ADHOC_DHCP, NMS.OPTION_ADHOC_INFRA_BEGIN, NMS.OPTION_ADHOC_SECTORED, NMS.OPTION_FORCED_ROOT, NMS.OPTION_IGMP, NMS.OPTION_LOCATION, NMS.OPTION_SIP

---

dynamicChannelAllocation

`public short` **`dynamicChannelAllocation`**

The node level flag determining whether downlink interfaces shall use the Dynamic Channel Allocation scheme.

A value of 0 will turn off the dynamic channel allocation scheme even if it is turned on for individual downlink interfaces.

---

countryCode

`public int` **`countryCode`**

The operating country code for the node.

A value of 0 indicates the default country code.

---

regulatoryDomain

`public int` **`regulatoryDomain`**

The operating regulatory domain for the node.

See Also:

NMS.REG_DOMAIN_CODE_NONE, NMS.REG_DOMAIN_CODE_CUSTOM, NMS.REG_DOMAIN_CODE_FCC, NMS.REG_DOMAIN_CODE_ETSI

---

dfsRequired

```
public short dfsRequired
```

Specifies whether Dynamics Frequency Selection and RADAR detection is required for the `regulatoryDomain`.

Constructor Detail

NMS.GeneralConfiguration

```
public NMS.GeneralConfiguration()
```

---

com.meshdynamics.api

Class NMS.Hashtable

```
java.lang.Object
  └ com.meshdynamics.api.NMS.Hashtable
```

Enclosing class:
NMS

```
public static class NMS.Hashtable
extends java.lang.Object
```

The Hashtable class provides an implementation of a Hashtable of generic 'Object' keys and generic 'Object' values.

Constructor Summary

NMS.Hashtable()	Default constructor.

Method Summary

void	**clear**() Clears the hashtable.
java.lang.Object	**get**(java.lang.Object key) Retrieves the value for the specified key.
java.util.Enumeration<java.lang.Object>	**keys**() Returns an Enumeration of all the keys in the hashtable.
void	**put**(java.lang.Object key, java.lang.Object value) Inserts the specified value for the specified key into the hashtable.
void	**remove**(java.lang.Object key) Removes the specified key from the hashtable.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

NMS.Hashtable

```
public NMS.Hashtable()
```

Default constructor.

Method Detail

get

```
public java.lang.Object get(java.lang.Object key)
```

Retrieves the value for the specified key.

Parameters:
`key` - the key for which the value is to be retrieved
Returns:
the value

---

put

```
public void put(java.lang.Object key,
                java.lang.Object value)
```

Inserts the specified value for the specified key into the hashtable.

Parameters:
`key` - the key for which the value is to be inserted
`value` - the value to be inserted

---

remove

```
public void remove(java.lang.Object key)
```

Removes the specified key from the hashtable.

---

clear

```
public void clear()
```

Clears the hashtable.

---

keys

```
public java.util.Enumeration<java.lang.Object> keys()
```

Returns an `Enumeration` of all the keys in the hashtable.

Returns:
`Enumeration` object for the keys

---

com.meshdynamics.api

Class NMS.InterfaceConfiguration

```
java.lang.Object
  └ com.meshdynamics.api.NMS.InterfaceConfiguration
```

Enclosing class:
NMS

```
public static class NMS.InterfaceConfiguration
extends java.lang.Object
```

Defines the interface level settings for a `NMS.Node`.

See Also:
`NMS.Node.getInterfaces()`, `NMS.Node.getInterfaceConfigurationByName(java.lang.String)`

Field Summary

int	**ackTimeout** The timeout in multiples of `10` micro-seconds for the 802.11 ACK frame.
short	**allowClientConnection** When set to `0`, does not allow standard 802.11 clients to connect to the interface, and only allows child mesh nodes to connect to the interface.
NMS.ShortArray	**dcaList** When `dynamicChannelAllocation` is non-zero, downlink interfaces choose the best channel from the integers specified in this array.
short	**dynamicChannelAllocation** When set to `0`, disables dynamic channel allocation and forces the interface to use the channel specified by `manualChannel`.
java.lang.String	**essid** The ESSID used in 802.11 beacons and 802.11 probe-response packets transmitted by the downlink interface.
int	**fragThreshold** The 802.11 fragmentation threshold for the interface.
short	**hideEssid** When non-zero causes the ESSID field of 802.11 beacons and broadcast probe-responses to contain an empty string.
short	**identifier** The identifier for the interface.
java.lang.String	**macAddress** The MAC address of the interface.
short	

	manualChannel The channel to be used when dynamicChannelAllocation is set to 0.
int	**maxTransmitRate** The maximum transmit rate for the interface.
java.lang.String	**name** The name of the interface.
short	**operatingChannel** The current operating channel for the interface.
short	**phySubType** Defines the physical layer sub-type used by the interface.
short	**phyType** Defines the Physical layer used by the interface.
int	**rtsThreshold** The 802.11 RTS threshold for the interface.
java.lang.Object	**securityInfo** Opaque object containing the security settings for the interface.
short	**securityType** The encryption/authentication scheme used to secure connections on the interface.
int	**transmitPower** The transmit power for the interface.
short	**usageType** Defines the role in which the interface is used during the node's operation.

Constructor Summary

NMS.InterfaceConfiguration() Default constructor.
NMS.InterfaceConfiguration(java.lang.String objectNotation) Initializes the configuration from the object notation string.

Method Summary

java.lang.String	**toObjectNotation**() Returns a string containing the object notation representation for the interface.
java.lang.String	**toString**()

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

name

`public java.lang.String` **`name`**

The name of the interface.

---

macAddress

`public java.lang.String` **`macAddress`**

The MAC address of the interface.

NOTE: The value of this field will be ignored in calls to `NMS.Node.setInterfaceConfiguration(com.meshdynamics.api.NMS.InterfaceConfiguration)`.

---

identifier

`public short` **`identifier`**

The identifier for the interface.

The interfaces of a node are identified according to the `usageType` and `phySubType` fields.

e.g. For a node with two 802.11a downlinks and a 802.11g downlink, the first downlink shall have an identifier of 0, while the 2nd will have 1.

The 802.11g downlink will have an identifier of 0.

NOTE: The value of this field will be ignored in calls to `NMS.Node.setInterfaceConfiguration(com.meshdynamics.api.NMS.InterfaceConfiguration)`.

---

usageType

`public short` **`usageType`**

Defines the role in which the interface is used during the node's operation.

NOTE: The value of this field will be ignored in calls to `NMS.Node.setInterfaceConfiguration(com.meshdynamics.api.NMS.InterfaceConfiguration)`.

See Also:

`NMS.USAGE_TYPE_DOWNLINK`, `NMS.USAGE_TYPE_UPLINK`, `NMS.USAGE_TYPE_SCANNER`

---

phyType

`public short` **`phyType`**

Defines the Physical layer used by the interface.

NOTE: The value of this field will be ignored in calls to `NMS.Node.setInterfaceConfiguration(com.meshdynamics.api.NMS.InterfaceConfiguration)`.

See Also:

NMS.PHY_TYPE_ETHERNET, NMS.PHY_TYPE_802_11

---

phySubType

```
public short phySubType
```

Defines the physical layer sub-type used by the interface.

See Also:

NMS.PHY_SUB_TYPE_IGNORE, NMS.PHY_SUB_TYPE_802_11_A, NMS.PHY_SUB_TYPE_802_11_B, NMS.PHY_SUB_TYPE_802_11_G, NMS.PHY_SUB_TYPE_802_11_BG, NMS.PHY_SUB_TYPE_802_11_PSQ, NMS.PHY_SUB_TYPE_802_11_PSH, NMS.PHY_SUB_TYPE_802_11_PSF

---

essid

```
public java.lang.String essid
```

The ESSID used in 802.11 beacons and 802.11 probe-response packets transmitted by the downlink interface.

This field is ignored for 802.11 uplink, scanner interfaces.

For ETHERNET downlinks, this field specifies the VLAN configuration for the ethernet port :

- ESSID of a VLAN - only allows the specified VLAN
- MD-PRIV-SSID-NO-VLAN - No VLANs allowed.
- Other - All VLANs allowed

---

hideEssid

```
public short hideEssid
```

When non-zero causes the ESSID field of 802.11 beacons and broadcast probe-responses to contain an empty string.

This field is ignored for 802.11 uplink, scanner interfaces and by all ethernet interfaces.

---

maxTransmitRate

```
public int maxTransmitRate
```

The maximum transmit rate for the interface.

When set to `0`, the interface uses all the transmit rates defined by the physical layer sub-type.

This field is ignored for ethernet interfaces.

---

transmitPower

```
public int transmitPower
```

The transmit power for the interface.

This field is ignored for ethernet interfaces.

---

ackTimeout

`public int` **`ackTimeout`**

The timeout in multiples of `10` micro-seconds for the 802.11 ACK frame.

Transmissions with the ACK frame not arriving within the ackTimeout value are considered erroneous and are retried.

This field is ignored for ethernet interfaces.

---

allowClientConnection

`public short` **`allowClientConnection`**

When set to `0`, does not allow standard 802.11 clients to connect to the interface, and only allows child mesh nodes to connect to the interface.

This field is ignored for ethernet interfaces.

---

fragThreshold

`public int` **`fragThreshold`**

The 802.11 fragmentation threshold for the interface.

All packets larger than the fragThreshold shall be fragmented.

This field is ignored for ethernet interfaces.

---

rtsThreshold

`public int` **`rtsThreshold`**

The 802.11 RTS threshold for the interface.

All packets larger than the rtsThreshold shall be preceeded by the standard 802.11 RTS/CTS mechanism to ensure error free reception.

This field is ignored for ethernet interfaces.

---

dynamicChannelAllocation

`public short` **`dynamicChannelAllocation`**

When set to `0`, disables dynamic channel allocation and forces the interface to use the channel specified by

`manualChannel`.

When set to a non-zero value, the interface chooses the best channel from the `dcaList` for operation.

This field is ignored for 802.11 uplink scanner and ethernet interfaces.

See Also:
`dcaList`, `manualChannel`

---

manualChannel

`public short` **`manualChannel`**

The channel to be used when `dynamicChannelAllocation` is set to 0.

This field is ignored for 802.11 uplink scanner and ethernet interfaces.

See Also:
`dynamicChannelAllocation`

---

dcaList

`public NMS.ShortArray` **`dcaList`**

When `dynamicChannelAllocation` is non-zero, downlink interfaces choose the best channel from the integers specified in this array.

For uplink interfaces, if the list is empty, all channels shall be scanned. If the list is non-empty only the channels specified in the list will be scanned for parent selection.

NOTE: The list must not be empty for uplink interfaces if the node is in disjoint-adhoc mode.

For scanner interfaces, the list determines the channels that will be scanned for detecting prospective parent nodes.

This field is ignored for ethernet interfaces.

See Also:
`dynamicChannelAllocation`

---

securityType

`public short` **`securityType`**

The encryption/authentication scheme used to secure connections on the interface.

This field is ignored for 802.11 uplink scanner and ethernet interfaces.

See Also:
`NMS.SECURITY_TYPE_NONE`, `NMS.SECURITY_TYPE_WEP_104`, `NMS.SECURITY_TYPE_WEP_40`, `NMS.SECURITY_TYPE_WPA2_ENTERPRISE`, `NMS.SECURITY_TYPE_WPA2_PERSONAL`, `NMS.SECURITY_TYPE_WPA_ENTERPRISE`, `NMS.SECURITY_TYPE_WPA_PERSONAL`

---

securityInfo

`public java.lang.Object` **`securityInfo`**

Opaque object containing the security settings for the interface.

The field represents a `NMS.WEPSecurity` object if `securityType` is `NMS.SECURITY_TYPE_WEP_104` or `NMS.SECURITY_TYPE_WEP_40`.

The field represents a `NMS.WPAPersonalSecurity` object if `securityType` is `NMS.SECURITY_TYPE_WPA2_PERSONAL` or `NMS.SECURITY_TYPE_WPA_PERSONAL`.

The field represents a `NMS.WPAEnterpriseSecurity` object if `securityType` is `NMS.SECURITY_TYPE_WPA2_ENTERPRISE` or `NMS.SECURITY_TYPE_WPA_ENTERPRISE`.

This field is ignored for 802.11 uplink scanner and ethernet interfaces.

See Also:
`securityType`, `NMS.WEPSecurity`, `NMS.WPAPersonalSecurity`, `NMS.WPAEnterpriseSecurity`

---

operatingChannel

`public short` **`operatingChannel`**

The current operating channel for the interface.

Constructor Detail

NMS.InterfaceConfiguration

`public` **`NMS.InterfaceConfiguration`**`()`

Default constructor.

NMS.InterfaceConfiguration

`public` **`NMS.InterfaceConfiguration`**`(java.lang.String objectNotation)`

Initializes the configuration from the object notation string.

Parameters:
`objectNotation` - the object notation string

Method Detail

toString

`public java.lang.String` **`toString`**`()`

Overrides:

`toString` in class `java.lang.Object`

---

toObjectNotation

`public java.lang.String` **`toObjectNotation`**`()`

Returns a string containing the object notation representation for the interface.

Returns:

string containing object notation representation of the interface

---

---

com.meshdynamics.api

Interface NMS.NeighborNode

Enclosing class:

NMS

`public static interface NMS.NeighborNode`

Defines the properties of all neighbor nodes detected by a `NMS.Node`

See Also:

`NMS.Node.getNeighborNodes()`

Method Summary

`int`	**`getDownlinkCount`**`()` Returns the number of downlink radios seen by the node.
`NMS.Node`	**`getNode`**`()` Returns a reference to the `NMS.Node` object representing the neighbor.
`int`	**`getUplinkSignal`**`()` Returns the RSSI of the neighbor's first downlink signal as seen by the uplink.
`int`	**`getUplinkSignal`**`(int downlinkIndex)` Returns the RSSI as seen by the uplink from the specific downlink of the neighbor.
`int`	**`getUplinkTxBitRate`**`()` Returns the transmit rate from the uplink to the neighbor's first downlink.
`int`	**`getUplinkTxBitRate`**`(int downlinkIndex)` Returns the transmit rate from the uplink to the specfic downlink of the neighbor.

Method Detail

getNode

`NMS.Node` **`getNode`**`()`

Returns a reference to the `NMS.Node` object representing the neighbor.

Returns:

a reference to the `NMS.Node` object representing the neighbor

---

getUplinkSignal

`int` **`getUplinkSignal`**`()`

Returns the RSSI of the neighbor's first downlink signal as seen by the uplink.

Returns:
signal RSSI
See Also:
`getUplinkSignal(int)`

---

getUplinkTxBitRate

`int` **`getUplinkTxBitRate`**`()`

Returns the transmit rate from the uplink to the neighbor's first downlink.

Returns:
transmit rate
See Also:
`getUplinkTxBitRate(int)`

---

getUplinkSignal

`int` **`getUplinkSignal`**`(int downlinkIndex)`

Returns the RSSI as seen by the uplink from the specific downlink of the neighbor.

Parameters:
`downlinkIndex` - the index of the neighbor's downlink
Returns:
signal RSSI

---

getUplinkTxBitRate

`int` **`getUplinkTxBitRate`**`(int downlinkIndex)`

Returns the transmit rate from the uplink to the specfic downlink of the neighbor.

Parameters:
`downlinkIndex` - the index of the neighbor's downlink
Returns:
transmit rate

---

getDownlinkCount

`int` **`getDownlinkCount`**`()`

Returns the number of downlink radios seen by the node.

Returns:
downlink count

---

com.meshdynamics.api

Interface NMS.Network

Enclosing class:
NMS

`public static interface NMS.Network`

The `Network` interface defines all properties and actions associated with a mesh network.

A mesh network is a community of mesh nodes that can :

- Communicate with each other using a common security parameters.
- Be managed as a single entity.

Method Summary

int	**addListener**(NMS.NetworkListener networklistener) Adds the specified NetworkListener callback hook to the mesh network.
int	**deleteNode**(NMS.Node node) Deletes the specified node from the mesh network.
java.lang.String	**getName**() Returns the name of the mesh network.
NMS.Node	**getNodeByMacAddress**(java.lang.String macAddress) Returns the Node object representing the specified MAC-address.
java.util.Enumeration<NMS.Node>	**getNodes**() Returns an Enumeration of all mesh nodes in the network.
int	**removeListener**(NMS.NetworkListener networklistener) Removes the specified NetworkListener callback hook from the mesh network.
int	**waitForNodeDetect**(java.lang.String macAddresses, long timeout) Blocks the calling thread until all the nodes specified in macAddresses parameter are fully detected and configurable.

Method Detail

getName

`java.lang.String **getName**()`

Returns the name of the mesh network.

Returns:
`String` object containing the name of the mesh network

---

getNodes

`java.util.Enumeration<NMS.Node> **getNodes**()`

Returns an `Enumeration` of all mesh nodes in the network.

Returns:
`Enumeration` of all mesh nodes in the network.
See Also:
`NMS.Node`

---

deleteNode

`int **deleteNode**(NMS.Node node)`

Deletes the specified node from the mesh network.

Parameters:
`node` - the node to be deleted
Returns:
`0` if successful

---

addListener

`int **addListener**(NMS.NetworkListener networklistener)`

Adds the specified `NetworkListener` callback hook to the mesh network.

The `NetworkListener` callback hook enables the caller to receive information on the events that occur in the mesh network.

Parameters:
`networklistener` - the `NetworkListener` callback hook to be added
Returns:
`0` if successful
See Also:
`NMS.NetworkListener`

---

removeListener

`int **removeListener**(NMS.NetworkListener networklistener)`

Removes the specified `NetworkListener` callback hook from the mesh network.

If successful, the caller will no longer be able to receive information on the events that occur in the mesh network.

Parameters:
`networklistener` - the `NetworkListener` callback hook to be removed
Returns:
`0` if successful
See Also:
`NMS.NetworkListener`

---

getNodeByMacAddress

`NMS.Node` **`getNodeByMacAddress`**`(java.lang.String macAddress)`

Returns the `Node` object representing the specified MAC-address.

Parameters:
`macAddress` - the mesh node's unit MAC-address to be searched
Returns:
`Node` object representing the specified MAC-address.
See Also:
`NMS.Node`

---

waitForNodeDetect

```
int waitForNodeDetect(java.lang.String macAddresses,
                      long timeout)
```

Blocks the calling thread until all the nodes specified in `macAddresses` parameter are fully detected and configurable.

Parameters:
`macAddresses` - A string containing comma-seperated list of MAC-addresses to detect
`timeout` - the number of milli-seconds to block until nodes get detected
Returns:
`0` if successful or negative integer if a timeout occurs.

---

com.meshdynamics.api

Interface NMS.NetworkListener

Enclosing class:
NMS

```
public static interface NMS.NetworkListener
```

The `NetworkListener` interface is used to receive events on a mesh network.

See Also:
`NMS.Network.addListener(com.meshdynamics.api.NMS.NetworkListener)`

Method Summary

int	**onEvent**(int event, NMS.Network network, NMS.Node node) This method is called when an event occurs on the network.

Method Detail

onEvent

```
int onEvent(int event,
            NMS.Network network,
            NMS.Node node)
```

This method is called when an event occurs on the network.

Parameters:

`event` - the code specifying the event that occurred. It can be one of the following:

`NMS.EVENT_NODE_DEAD`,`NMS.EVENT_NODE_HEARTBEAT`,
`NMS.EVENT_NODE_HEARTBEAT_MISS`,`NMS.EVENT_NODE_SCAN`

`network` - the network on which the event occurred
`node` - the node for which the event occurred

Returns:

Currently the return value is ignored and must be set to 0

---

com.meshdynamics.api

Interface NMS.Node

Enclosing class:
NMS

public static interface **NMS.Node**

The Node interface defines all the properties and actions that can be carried out on a mesh node.

Method Summary

int	**addVlan**(NMS.VlanConfiguration configuration) Adds the specified VLAN to the Node.
int	**beginConfigurationUpdate**() Starts a configuration transaction bracket.
int	**cancelConfigurationUpdate**() Closes the current configuration transaction bracket without sending the configuration update.
int	**commitConfigurationUpdate**() Closes the current configuration transaction bracket and sends the updated configuration to the Node.
java.lang.String	**executeCommand**(java.lang.String command) Executes a Meshdynamics MeshCommand™ on the Node.
java.lang.String	**generateConfigMacro**(java.lang.String scriptLanguage) Generates a configuration macro script for the Node.
NMS.ACLConfiguration	**getACLConfiguration**() Returns the Access Control List configuration for the Node.
java.util.Enumeration<NMS.ConnectedDevice>	**getConnectedDevices**() Returns an Enumeration of devices that are connected to this Node.
short	**getCpuUsage**() Returns the current average CPU usage for the node.
NMS.EffistreamRule	**getEffistreamRules**() Returns the Effistream™ rule hierarchy for the Node.
short	**getFirmwareVersionMajor**() Returns the major firmware version for the Node.
short	**getFirmwareVersionMinor**() Returns the minor firmware version for the Node.
short	**getFirmwareVersionVariant**() Returns the firmware version variant for the Node.
short	**getFreeRAM**() Returns the amount of free RAM in Mega-bytes.
NMS.GeneralConfiguration	**getGeneralConfiguration**() Returns the node level configuration of the Node.
short	**getGpsAltitude**() Returns the current operational altitude in meters.
java.lang.String	

	getGpsCurrentLatitude() Returns the current operational latitude coordinate in decimal format.
java.lang.String	**getGpsCurrentLongitude**() Returns the current operational longitude coordinate in decimal format.
short	**getGpsSpeed**() Returns the current operational speed in Km/Hr.
long	**getHeartbeatSqnr**() Returns the sequence number of the last heartbeat received from the node.
short	**getHopCount**() Returns the current hop level for the node.
short	**getInputVoltage**() Returns the current input voltage to the node.
NMS.InterfaceConfiguration	**getInterfaceConfigurationByName**(java.lang.String name) Returns the configuration of the specified interface.
java.util.Enumeration<NMS.InterfaceConfiguration>	**getInterfaces**() Returns an Enumeration of all interfaces in the Node.
java.util.Enumeration<NMS.NeighborNode>	**getNeighborNodes**() Returns an Enumeration of nodes that this Node sees as neighbors.
java.lang.String	**getParentBssid**() Returns the MAC-address of the parent's downlink on which this Node is connected.
int	**getParentDownlinkSignal**() Returns the signal RSSI in packets received by the parent's downlink interface from this node's uplink.
int	**getParentDownlinkTxBitRate**() Returns the transmit rate used by the parent for packet's transmitted to this Node.
short	**getTemperature**() Returns the current node enclosure temperature.
short	**getTreeLinkRate**() Returns the 'Tree Link Rate' for the node.
java.lang.String	**getUnitMacAddress**() Returns the MAC address of the node formatted as a string.
NMS.VlanConfiguration	**getVlanConfigurationByTag**(short tag) Returns the configuration of the specified VLAN.
java.util.Enumeration<NMS.VlanConfiguration>	**getVlans**() Returns an Enumeration of all VLANS in the Node.
short	**isIpReachable**() Returns non-zero if this Node can be communicated with using IP.
boolean	**isMobile**() Returns whether the node is mobile or stationary.
boolean	**isRemote**() Returns whether the remote or local.
void	**reboot**() REBOOT's the Node.
short	**rebootRequired**() Returns non-zero if a 'REBOOT' is required for the Node.
int	**removeVlan**(short tag) Removes the specified VLAN from the Node.
int	**restoreDefaults**() Restore's the Node to factory configuration.
java.lang.String	**runPerformanceTest**(int recordCount, short type, short protocol,

	int udpBandWidth) Provides network performance information to the Node.
int	**setACLConfiguration**(NMS.ACLConfiguration configuration) Sets the Node's Access Control List configuration.
int	**setEffistreamRules**(NMS.EffistreamRule rules) Updates the Effistream™ rule hierarchy for the Node.
int	**setGeneralConfiguration**(NMS.GeneralConfiguration configuration) Updates the node level configuration for the Node.
int	**setInterfaceConfiguration**(NMS.InterfaceConfiguration configuration) Updates the interface configuration for the Node.
int	**setVlanConfiguration**(NMS.VlanConfiguration configuration) Sets the configuration of an existing VLAN in the Node.
int	**setVlans**(NMS.ObjectArray vlans) Sets the Node's VLAN list from a ObjectArray.
int	**upgradeFirmware**(java.lang.String firmwareFilePath) Upgrades the firmware of the Node.

Method Detail

getUnitMacAddress

java.lang.String **getUnitMacAddress**()

Returns the MAC address of the node formatted as a string.

Returns:
MAC address

---

getHeartbeatSqnr

long **getHeartbeatSqnr**()

Returns the sequence number of the last heartbeat received from the node.

Returns:
heartbeat sequence number

---

isMobile

boolean **isMobile**()

Returns whether the node is mobile or stationary.

Returns:
true if the node is mobile, false otherwise

---

isRemote

boolean **isRemote**()

Returns whether the remote or local.

Returns:
true if node is remote, false otherwise

---

getFreeRAM

```
short getFreeRAM()
```

Returns the amount of free RAM in Mega-bytes.

Returns:
free RAM in Mega-bytes

---

getInputVoltage

```
short getInputVoltage()
```

Returns the current input voltage to the node.

Returns:
node input voltage

---

getTreeLinkRate

```
short getTreeLinkRate()
```

Returns the 'Tree Link Rate' for the node.

The 'Tree Link Rate' is the lowest rate in the path from the node to the ROOT.

Returns:
the 'Tree Link Rate'

---

getHopCount

```
short getHopCount()
```

Returns the current hop level for the node.

Returns:
the number of hops away from the ROOT.

---

getCpuUsage

```
short getCpuUsage()
```

Returns the current average CPU usage for the node.

Returns:
the average cpu usage as a percentage

---

getTemperature

```
short getTemperature()
```

Returns the current node enclosure temperature.

Returns:
the current temperature inside the node enclosure in Celcius.

---

getParentDownlinkSignal

`int` **`getParentDownlinkSignal`**`()`

Returns the signal RSSI in packets received by the parent's downlink interface from this node's uplink.

Returns:
the signal RSSI received by the parent's downlink interface.

---

getParentDownlinkTxBitRate

`int` **`getParentDownlinkTxBitRate`**`()`

Returns the transmit rate used by the parent for packet's transmitted to this `Node`.

Returns:
the transmit rate for packets transmitted by parent's downlink.

---

getParentBssid

`java.lang.String` **`getParentBssid`**`()`

Returns the MAC-address of the parent's downlink on which this `Node` is connected.

Returns:
MAC-address of parent's downlink interface

---

getGpsCurrentLatitude

`java.lang.String` **`getGpsCurrentLatitude`**`()`

Returns the current operational latitude coordinate in decimal format.

Coordinates South of the equator are represented by a negative number.

Returns:
the current operational latitude coordinate

---

getGpsCurrentLongitude

`java.lang.String` **`getGpsCurrentLongitude`**`()`

Returns the current operational longitude coordinate in decimal format.

Coordinates West of the meridian are represented by a negative number.

Returns:
the current operational longitude coordinate

---

getGpsSpeed

`short` **`getGpsSpeed`**`()`

Returns the current operational speed in Km/Hr.

Returns:
the current operational speed

---

getGpsAltitude

`short` **`getGpsAltitude`**`()`

Returns the current operational altitude in meters.

Returns:
the the current operational altitude in meters

---

getFirmwareVersionMajor

`short` **`getFirmwareVersionMajor`**`()`

Returns the major firmware version for the `Node`.

Returns:
the major firmware version.

---

getFirmwareVersionMinor

`short` **`getFirmwareVersionMinor`**`()`

Returns the minor firmware version for the `Node`.

Returns:
the minor firmware version.

---

getFirmwareVersionVariant

`short` **`getFirmwareVersionVariant`**`()`

Returns the firmware version variant for the `Node`.

Returns:
the firmware version variant.

---

isIpReachable

`short` **`isIpReachable`**`()`

Returns non-zero if this `Node` can be communicated with using IP.

Returns:
`0` if node is not IP-reachable.
See Also:
`NMS.GeneralConfiguration.ipAddress`

---

rebootRequired

`short` **`rebootRequired`**`()`

Returns non-zero if a 'REBOOT' is required for the `Node`.

Returns:
`0` if the changes to the `Node`'s configuration dot not require a reboot. `non-zero` if a reboot is required.

---

getNeighborNodes

`java.util.Enumeration<NMS.NeighborNode>` **`getNeighborNodes`**`()`

Returns an `Enumeration` of nodes that this `Node` sees as neighbors.

Neighbor nodes are pottential parent nodes, and are connected to, in the event of a link failure.

Returns:
`Enumeration` of `NeighborNode` objects

---

getConnectedDevices

`java.util.Enumeration<NMS.ConnectedDevice>` **`getConnectedDevices`**`()`

Returns an `Enumeration` of devices that are connected to this `Node`.

This method returns standard client devices and child mesh nodes.

Returns:
`Enumeration` of `ConnectedDevice` objects

---

getGeneralConfiguration

`NMS.GeneralConfiguration` **`getGeneralConfiguration`**`()`

Returns the node level configuration of the `Node`.

Returns:
the node level configuration of the `Node`

---

getInterfaces

`java.util.Enumeration<NMS.InterfaceConfiguration>` **`getInterfaces`**`()`

Returns an `Enumeration` of all interfaces in the `Node`.

Returns:
`Enumeration` of `InterfaceConfiguration` objects

---

getVlans

`java.util.Enumeration<NMS.VlanConfiguration>` **`getVlans`**`()`

Returns an `Enumeration` of all VLANS in the `Node`.

Returns:
`Enumeration` of `VlanConfiguration` objects

---

getInterfaceConfigurationByName

`NMS.InterfaceConfiguration` **`getInterfaceConfigurationByName`**`(java.lang.String name)`

Returns the configuration of the specified interface.

Parameters:
`name` - the name of the interface
Returns:
`InterfaceConfiguration` object for the interface

---

getVlanConfigurationByTag

`NMS.VlanConfiguration` **`getVlanConfigurationByTag`**`(short tag)`

Returns the configuration of the specified VLAN.

Parameters:
`tag` - the VLAN identifier
Returns:
`VlanConfiguration` object for the VLAN

---

getEffistreamRules

`NMS.EffistreamRule` **`getEffistreamRules`**`()`

Returns the Effistream™ rule hierarchy for the `Node`.

Returns:
`EffistreamRule` object hierachy

---

getACLConfiguration

`NMS.ACLConfiguration` **`getACLConfiguration`**`()`

Returns the Access Control List configuration for the `Node`.

Returns:
`ACLConfiguration` object

---

reboot

`void` **`reboot`**`()`

REBOOT's the `Node`.

---

restoreDefaults

`int` **`restoreDefaults`**`()`

Restore's the `Node` to factory configuration.

Returns:
`0` on success

---

executeCommand

`java.lang.String` **`executeCommand`**`(java.lang.String command)`

Executes a Meshdynamics MeshCommand™ on the `Node`.

Parameters:
`command` - the Meshdynamics MeshCommand™ to execute
Returns:
the result of the command

---

upgradeFirmware

```
int upgradeFirmware(java.lang.String firmwareFilePath)
```

Upgrades the firmware of the `Node`.

The firmware file must be one that is created specifically for the MAC address of the `Node`.

Parameters:
`firmwareFilePath` - the path to the firmware upgrade file.
Returns:
`0` on success

---

runPerformanceTest

```
java.lang.String runPerformanceTest(int recordCount,
                                    short type,
                                    short protocol,
                                    int udpBandWidth)
```

Provides network performance information to the `Node`.

The performance test is run from the host to the `Node` and hence will reflect the network performance of all links along the path.

Parameters:
`recordCount` - the number of performanc records to be run
`type` - the type of the performance run, can be one of `NMS.PERFORMANCE_TYPE_SINGLE`, `NMS.PERFORMANCE_TYPE_DUAL_INDIVIDUAL`, `NMS.PERFORMANCE_TYPE_DUAL_SIMULTANEOUS`
`protocol` - the protocol to be used, can be one of `NMS.PERFORMANCE_PROTOCOL_TCP`, `NMS.PERFORMANCE_PROTOCOL_UDP`.
`udpBandWidth` - when using `PERFORMANCE_PROTOCOL_UDP`, the bandwidth in Kbps.
Returns:
the result of the performance test

---

setGeneralConfiguration

```
int setGeneralConfiguration(NMS.GeneralConfiguration configuration)
```

Updates the node level configuration for the `Node`.

If `beginConfigurationUpdate` has been called prior to this method, the updated configuration will be sent upon a call to the method `commitConfigurationUpdate`.

If `beginConfigurationUpdate` has not been called prior to this method, the configuration is sent immedietly.

Parameters:
`configuration` - the node level configuration
Returns:
`0` upon success

---

setInterfaceConfiguration

```
int setInterfaceConfiguration(NMS.InterfaceConfiguration configuration)
```

Updates the interface configuration for the `Node`.

The interface is specified by the `name` field of the `InterfaceConfiguration` object.

If `beginConfigurationUpdate` has been called prior to this method, the updated configuration will be sent upon a call to the method `commitConfigurationUpdate`.

If `beginConfigurationUpdate` has not been called prior to this method, the configuration is sent immedietly.

Parameters:
`configuration` - the configuration for the interface

Returns:
`0` upon success

---

setEffistreamRules

```
int setEffistreamRules(NMS.EffistreamRule rules)
```

Updates the Effistream™ rule hierarchy for the `Node`.

If `beginConfigurationUpdate` has been called prior to this method, the updated configuration will be sent upon a call to the method `commitConfigurationUpdate`.

If `beginConfigurationUpdate` has not been called prior to this method, the configuration is sent immedietly.

Parameters:
`rules` - the Effistream™ rule hierarchy
Returns:
`0` upon success

---

addVlan

```
int addVlan(NMS.VlanConfiguration configuration)
```

Adds the specified VLAN to the `Node`.

If `beginConfigurationUpdate` has been called prior to this method, the updated configuration will be sent upon a call to the method `commitConfigurationUpdate`.

If `beginConfigurationUpdate` has not been called prior to this method, the configuration is sent immedietly.

Parameters:
`configuration` - the `VlanConfiguration` object
Returns:
`0` upon success

---

setVlanConfiguration

```
int setVlanConfiguration(NMS.VlanConfiguration configuration)
```

Sets the configuration of an existing VLAN in the `Node`.

The `essid` and `tag` fields of the `VlanConfiguration` object are used to identify the existing VLAN.

If no existing VLAN exisits, the method returns an error.

If `beginConfigurationUpdate` has been called prior to this method, the updated configuration will be sent upon a call to the method `commitConfigurationUpdate`.

If `beginConfigurationUpdate` has not been called prior to this method, the configuration is sent immedietly.

Parameters:
`configuration` - the `VlanConfiguration` object
Returns:
`0` upon success

---

removeVlan

```
int removeVlan(short tag)
```

Removes the specified VLAN from the `Node`.

The tag field is used to identify the VLAN.

If no existing VLAN exisits, the method returns an error.

If `beginConfigurationUpdate` has been called prior to this method, the updated configuration will be sent upon a call to the method `commitConfigurationUpdate`.

If `beginConfigurationUpdate` has not been called prior to this method, the configuration is sent immedietly.

Parameters:
`tag` - the tag to identify the existing VLAN
Returns:
`0` upon success

---

setVlans

`int` **`setVlans`**`(`NMS.ObjectArray `vlans)`

Sets the `Node`'s VLAN list from a `ObjectArray`.

This method delete's all existing VLANs and adds all VLANs in the `ObjectArray`.

If `beginConfigurationUpdate` has been called prior to this method, the updated configuration will be sent upon a call to the method `commitConfigurationUpdate`.

If `beginConfigurationUpdate` has not been called prior to this method, the configuration is sent immedietly.

Parameters:
`vlans` - `ObjectArray` containing `VlanConfiguration` objects
Returns:
`0` upon success

---

setACLConfiguration

`int` **`setACLConfiguration`**`(`NMS.ACLConfiguration `configuration)`

Sets the `Node`'s Access Control List configuration.

This method delete's all existing entries from the ACL configuration and sets the `Node`'s Access Control List configuration as specified by the `ACLConfiguration` object.

If `beginConfigurationUpdate` has been called prior to this method, the updated configuration will be sent upon a call to the method `commitConfigurationUpdate`.

If `beginConfigurationUpdate` has not been called prior to this method, the configuration is sent immedietly.

Parameters:
`configuration` - the `ACLConfiguration` object
Returns:
`0` upon success

---

generateConfigMacro

`java.lang.String` **`generateConfigMacro`**`(java.lang.String scriptLanguage)`

Generates a configuration macro script for the `Node`.

Parameters:
`scriptLanguage` - the scripting lanugage to use

Returns:
string containing the configuration macro script

---

beginConfigurationUpdate

`int` **`beginConfigurationUpdate`**`()`

Starts a configuration transaction bracket.

After a call to this method, calls that update the `Node`'s configuration are be not be sent immedietly, but are deferred until a call to `commitConfigurationUpdate`.

The configuration transaction bracket can be closed by a call to `commitConfigurationUpdate` or to `cancelConfigurationUpdate`.

Returns:
`0` upon success

---

cancelConfigurationUpdate

`int` **`cancelConfigurationUpdate`**`()`

Closes the current configuration transaction bracket without sending the configuration update.

Returns:
`0` upon success

---

commitConfigurationUpdate

`int` **`commitConfigurationUpdate`**`()`

Closes the current configuration transaction bracket and sends the updated configuration to the `Node`.

Returns:
`0` upon success

---

com.meshdynamics.api

Class NMS.ObjectArray

```
java.lang.Object
  └ com.meshdynamics.api.NMS.ObjectArray
```

Enclosing class:
NMS

```
public static class NMS.ObjectArray
extends java.lang.Object
```

The ObjectArray class provides an interface to a growable array that stores object references.

Constructor Summary

`NMS.ObjectArray()`	Default constructor to create the array with 0 elements.
`NMS.ObjectArray(int length)`	Constructor to create the array with specified number of elements initialized to null.

Method Summary

`void`	`add(java.lang.Object value)` Add a object reference to the end of the array and increase the length by 1.
`void`	`clear()` Removes all elements in the array and sets the number of elements to 0.
`java.lang.Object`	`get(int index)` Retrieves the object reference at the specified index.
`int`	`length()` Retrieve the number of elements in the `ObjectArray`.
`void`	`removeAt(int index)` Removes the element at the specified index.
`void`	`set(int index, java.lang.Object value)` Set the object reference at the specified index.
`java.lang.String`	`toObjectNotation()` Returns a string containing the object notation representation for the `ObjectArray`.
`java.lang.String`	`toString()`

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

NMS.ObjectArray

```
public NMS.ObjectArray()
```

Default constructor to create the array with 0 elements.

NMS.ObjectArray

```
public NMS.ObjectArray(int length)
```

Constructor to create the array with specified number of elements initialized to null.

Method Detail

set

```
public void set(int index,
                java.lang.Object value)
```

Set the object reference at the specified index.

Parameters:
`index` - the index
`value` - the object reference

---

get

```
public java.lang.Object get(int index)
```

Retrieves the object reference at the specified index.

Parameters:
`index` - the index
Returns:
the object reference

---

length

```
public int length()
```

Retrieve the number of elements in the `ObjectArray`.

Returns:
the number of elements

---

removeAt

```
public void removeAt(int index)
```

Removes the element at the specified index.

Parameters:
`index` - the index of the element to be removed.

---

add

```
public void add(java.lang.Object value)
```

Add a object reference to the end of the array and increase the length by 1.

Parameters:
`value` - the object reference to be added

---

clear

```
public void clear()
```

Removes all elements in the array and sets the number of elements to 0.

---

toString

```
public java.lang.String toString()
```

Overrides:
`toString` in class `java.lang.Object`

---

toObjectNotation

```
public java.lang.String toObjectNotation()
```

Returns a string containing the object notation representation for the `ObjectArray`.

Returns:
string containing object notation

---

com.meshdynamics.api

Class NMS.ShortArray

```
java.lang.Object
  └ com.meshdynamics.api.NMS.ShortArray
```

Enclosing class:
NMS

```
public static class NMS.ShortArray
extends java.lang.Object
```

Defines an array of short integers.

Constructor Summary

NMS.ShortArray(int length) Constructs ShortArray object with specified number of elements.
NMS.ShortArray(short... numbers) Constructs ShortArray object with the specified elements.
NMS.ShortArray(java.lang.String values) Constructs ShortArray object from a comma seperated list of numbers.

Method Summary

short	**get**(int index) Retrieve the value at the specified index.
int	**length**() Retrieve the number of elements in the ShortArray.
void	**set**(int index, short value) Set the value at specified index.
void	**set**(short... numbers) Set the elements of the ShortArray to the specified variable argument list of numbers.
void	**set**(java.lang.String values) Set the elements of the ShortArray from a comma seperated list of numbers.
java.lang.String	**toObjectNotation**() Returns a string containing the object notation representation for the ShortArray.
java.lang.String	**toString**()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

NMS.ShortArray

```
public NMS.ShortArray(int length)
```

Constructs ShortArray object with specified number of elements.

Parameters:
length - the number of elements

NMS.ShortArray

```
public NMS.ShortArray(short... numbers)
```

Constructs ShortArray object with the specified elements.

Parameters:
numbers - variable argument list of short inetegers

NMS.ShortArray

```
public NMS.ShortArray(java.lang.String values)
```

Constructs ShortArray object from a comma seperated list of numbers.

Parameters:
values - string containing comma seperated list of numbers

Method Detail

set

```
public void set(short... numbers)
```

Set the elements of the ShortArray to the specified variable argument list of numbers.

Parameters:
numbers - variable argument list of short inetegers

---

set

```
public void set(java.lang.String values)
```

Set the elements of the ShortArray from a comma seperated list of numbers.

Parameters:

`values` - string specifying comma seperated list of values

---

set

```
public void set(int index,
                short value)
```

Set the value at specified index.

Parameters:
`index` - the index
`value` - the value

---

get

```
public short get(int index)
```

Retrieve the value at the specified index.

Parameters:
`index` - the index
Returns:
the value at the specified index

---

length

```
public int length()
```

Retrieve the number of elements in the `ShortArray`.

Returns:
the number of elements

---

toString

```
public java.lang.String toString()
```

Overrides:
`toString` in class `java.lang.Object`

---

toObjectNotation

```
public java.lang.String toObjectNotation()
```

Returns a string containing the object notation representation for the `ShortArray`.

Returns:
string containing object notation representation

---

com.meshdynamics.api

Class NMS.Thread

```
java.lang.Object
  └java.lang.Thread
      └com.meshdynamics.api.NMS.Thread
```

All Implemented Interfaces:

java.lang.Runnable

Enclosing class:

NMS

```
public static class NMS.Thread
extends java.lang.Thread
```

The `Thread` class provides multi-threading functionality to scripting platforms.

Nested Class Summary

static interface	**NMS.Thread.Runnable** The `Runnable` interface is implemented by any class whose instances are executed by a thread.

Nested classes/interfaces inherited from class java.lang.Thread
java.lang.Thread.State, java.lang.Thread.UncaughtExceptionHandler

Field Summary

Fields inherited from class java.lang.Thread
MAX_PRIORITY, MIN_PRIORITY, NORM_PRIORITY

Constructor Summary

NMS.Thread(NMS.Thread.Runnable runnable) Default constructor

Method Summary

void	**run**()
static void	**sleep**(long milliSeconds)

	The `sleep` method blocks the calling thread for the specified number of milli-seconds. Since it is a static method, the calling thread does not have to be an instance of the `NMS.Thread` class.
`void`	**`start`**`()` Starts the thread.

Methods inherited from class java.lang.Thread
`activeCount, checkAccess, countStackFrames, currentThread, destroy, dumpStack, enumerate, getAllStackTraces, getContextClassLoader, getDefaultUncaughtExceptionHandler, getId, getName, getPriority, getStackTrace, getState, getThreadGroup, getUncaughtExceptionHandler, holdsLock, interrupt, interrupted, isAlive, isDaemon, isInterrupted, join, join, join, resume, setContextClassLoader, setDaemon, setDefaultUncaughtExceptionHandler, setName, setPriority, setUncaughtExceptionHandler, sleep, stop, stop, suspend, toString, yield`

Methods inherited from class java.lang.Object
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructor Detail

NMS.Thread

`public` **`NMS.Thread`**`(`[`NMS.Thread.Runnable`]` runnable)`

Default constructor

Parameters:

`runnable` - the reference to an object implementing the `Runnable` interface

Method Detail

sleep

`public static void` **`sleep`**`(long milliSeconds)`

The `sleep` method blocks the calling thread for the specified number of milli-seconds.

Since it is a static method, the calling thread does not have to be an instance of the `NMS.Thread` class.

Parameters:

`milliSeconds` - the number of milli-seconds to block

---

start

`public void` **`start`**`()`

Starts the thread.

Overrides:

`start` in class `java.lang.Thread`

---

run

```
public void run()
```

Specified by:
`run` in interface `java.lang.Runnable`

Overrides:
`run` in class `java.lang.Thread`

Package **Class** **Tree** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

---

com.meshdynamics.api

Interface NMS.Thread.Runnable

Enclosing class:
NMS.Thread

public static interface **NMS.Thread.Runnable**

The Runnable interface is implemented by any class whose instances are executed by a thread.

The interface defines a single method run that represents the running thread.

See Also:
NMS.Thread

Method Summary

void	**run**() The run method implements the logic for the thread.

Method Detail

run

void **run**()

The run method implements the logic for the thread.

com.meshdynamics.api

Class NMS.VlanConfiguration

```
java.lang.Object
  └ com.meshdynamics.api.NMS.VlanConfiguration
```

Enclosing class:
NMS

```
public static class NMS.VlanConfiguration
extends java.lang.Object
```

Defines the settings for a Virtual-LAN in a `NMS.Node`.

Field Summary

short	**dot11eCategory** The IEEE 802.11e access category to be used for packets for the VLAN.
short	**dot11eEnabled** Non-zero if IEEE 802.11e based QOS is enabled for the VLAN.
short	**dot1pPriority** The IEEE 802.1p bridge priority for the VLAN.
java.lang.String	**essid** The ESSID used in 802.11 probe-response packets.
java.lang.String	**name** The friendly name for the VLAN.
java.lang.Object	**securityInfo** Opaque object containing the security settings for the VLAN.
short	**securityType** The encryption/authentication scheme used to secure connections on the VLAN.
short	**tag** The IEEE 802.1q tag for the VLAN.

Constructor Summary

NMS.VlanConfiguration() Default constructor.
NMS.VlanConfiguration(java.lang.String objectNotation) Creates a VlanConfiguration object from a object notation string.

Method Summary

java.lang.String	**toObjectNotation**() Returns a string containing the object notation representation of the VlanConfiguration object.
java.lang.String	**toString**()

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

name

public java.lang.String **name**

The friendly name for the VLAN.

---

essid

public java.lang.String **essid**

The ESSID used in 802.11 probe-response packets.

---

tag

public short **tag**

The IEEE 802.1q tag for the VLAN.

---

dot11eEnabled

public short **dot11eEnabled**

Non-zero if IEEE 802.11e based QOS is enabled for the VLAN.

---

dot11eCategory

public short **dot11eCategory**

The IEEE 802.11e access category to be used for packets for the VLAN.

Ignored if `dot11eEnabled` is `0`.

---

dot1pPriority

public short **dot1pPriority**

The IEEE 802.1p bridge priority for the VLAN.

---

securityType

`public short` **`securityType`**

The encryption/authentication scheme used to secure connections on the VLAN.

See Also:
`NMS.SECURITY_TYPE_NONE`, `NMS.SECURITY_TYPE_WEP_104`, `NMS.SECURITY_TYPE_WEP_40`, `NMS.SECURITY_TYPE_WPA2_ENTERPRISE`, `NMS.SECURITY_TYPE_WPA2_PERSONAL`, `NMS.SECURITY_TYPE_WPA_ENTERPRISE`, `NMS.SECURITY_TYPE_WPA_PERSONAL`

---

securityInfo

`public java.lang.Object` **`securityInfo`**

Opaque object containing the security settings for the VLAN.

The field represents a `NMS.WEPSecurity` object if `securityType` is `NMS.SECURITY_TYPE_WEP_104` or `NMS.SECURITY_TYPE_WEP_40`.

The field represents a `NMS.WPAPersonalSecurity` object if `securityType` is `NMS.SECURITY_TYPE_WPA2_PERSONAL` or `NMS.SECURITY_TYPE_WPA_PERSONAL`.

The field represents a `NMS.WPAEnterpriseSecurity` object if `securityType` is `NMS.SECURITY_TYPE_WPA2_ENTERPRISE` or `NMS.SECURITY_TYPE_WPA_ENTERPRISE`.

This field is ignored for 802.11 uplink scanner and ethernet interfaces.

See Also:
`securityType`, `NMS.WEPSecurity`, `NMS.WPAPersonalSecurity`, `NMS.WPAEnterpriseSecurity`

Constructor Detail

NMS.VlanConfiguration

`public` **`NMS.VlanConfiguration`**`()`

Default constructor.

NMS.VlanConfiguration

`public` **`NMS.VlanConfiguration`**`(java.lang.String objectNotation)`

Creates a VlanConfiguration object from a object notation string.

Parameters:
`objectNotation` - the object notation string

Method Detail

toString

```
public java.lang.String toString()
```

Overrides:

`toString` in class `java.lang.Object`

---

toObjectNotation

```
public java.lang.String toObjectNotation()
```

Returns a string containing the object notation representation of the VlanConfiguration object.

Returns:

the object notation string

---

com.meshdynamics.api

Class NMS.WEPSecurity

```
java.lang.Object
  └ com.meshdynamics.api.NMS.WEPSecurity
```

Enclosing class:
NMS

```
public static class NMS.WEPSecurity
extends java.lang.Object
```

Defines the information used by the IEEE 802.11 **Wired Equivalent Privacy** (WEP) setting by a Node's downlink interface.

See Also:
`NMS.InterfaceConfiguration.securityType`, `NMS.InterfaceConfiguration.securityInfo`

Field Summary

short	**keyIndex** The index of the key used for transmitting packets.
NMS.ObjectArray	**wepKeys** An array of upto 4 WEP keys formatted as hexadecimal strings.

Constructor Summary

NMS.WEPSecurity() Default constructor.

Method Summary

java.lang.String	**toObjectNotation**() Returns a string containing the object notation representation of the `WEPSecurity` object
java.lang.String	**toString**()

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

keyIndex

```
public short keyIndex
```

The index of the key used for transmitting packets.

For WEP-40 the valid values are 0-3.

For WEP-104 the value is ignored.

---

wepKeys

```
public NMS.ObjectArray wepKeys
```

An array of upto 4 WEP keys formatted as hexadecimal strings.

When using WEP-40 the array shall contain 4 entries of 10 hexadecimal digits.

For WEP-104 the array shall contain 1 entry of 26 hexadecimal digits

Constructor Detail

NMS.WEPSecurity

```
public NMS.WEPSecurity()
```

Default constructor.

Method Detail

toString

```
public java.lang.String toString()
```

Overrides:
`toString` in class `java.lang.Object`

---

toObjectNotation

```
public java.lang.String toObjectNotation()
```

Returns a string containing the object notation representation of the `WEPSecurity` object

Returns:
the object notation string

---

com.meshdynamics.api

Class NMS.WPAEnterpriseSecurity

```
java.lang.Object
  └ com.meshdynamics.api.NMS.WPAEnterpriseSecurity
```

Enclosing class:
NMS

```
public static class NMS.WPAEnterpriseSecurity
extends java.lang.Object
```

Defines the information used for the Wifi Protected Access security setting by a Node's downlink interface in an enterprise environment.

See Also:
`NMS.InterfaceConfiguration.securityType`, `NMS.InterfaceConfiguration.securityInfo`

Field Summary

short	**cipherType** Defines the encryption mechanism to be used.
java.lang.String	**radiusServerIp** IP-address of the RADIUS server
short	**radiusServerPort** The UDP port used by the RADIUS server
java.lang.String	**radiusServerSecret** The secret key used to authenticate RADIUS packets sent by the node

Constructor Summary

NMS.WPAEnterpriseSecurity() Default constructor

Method Summary

java.lang.String	**toObjectNotation**() Returns a string containing the object notation representation of the WPAEnterpriseSecurity object.
java.lang.String	**toString**()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

radiusServerIp

public java.lang.String **radiusServerIp**

IP-address of the RADIUS server

---

radiusServerPort

public short **radiusServerPort**

The UDP port used by the RADIUS server

---

radiusServerSecret

public java.lang.String **radiusServerSecret**

The secret key used to authenticate RADIUS packets sent by the node

---

cipherType

public short **cipherType**

Defines the encryption mechanism to be used.

See Also:
NMS.CIPHER_CCMP, NMS.CIPHER_TKIP

Constructor Detail

NMS.WPAEnterpriseSecurity

public **NMS.WPAEnterpriseSecurity**()

Default constructor

Method Detail

toString

public java.lang.String **toString**()

Overrides:
toString in class java.lang.Object

toObjectNotation

`public java.lang.String` **`toObjectNotation`**`()`

Returns a string containing the object notation representation of the `WPAEnterpriseSecurity` object.

Returns:

the object notation string

com.meshdynamics.api

Class NMS.WPAPersonalSecurity

```
java.lang.Object
  └ com.meshdynamics.api.NMS.WPAPersonalSecurity
```

Enclosing class:
NMS

```
public static class NMS.WPAPersonalSecurity
extends java.lang.Object
```

Defines the information used for the Wifi Protected Access (WPA) security setting by a node's downlink interface.

See Also:
`NMS.InterfaceConfiguration.securityType`, `NMS.InterfaceConfiguration.securityInfo`

Field Summary

short	**cipherType** Defines the encryption mechanism to be used.
java.lang.String	**preSharedKey** The 256-bit pre-shared key (PSK) formatted as a hexidecimal string.

Constructor Summary

NMS.WPAPersonalSecurity() Default constructor

Method Summary

java.lang.String	**toObjectNotation**() Returns a string containing the object notation representation of the `WPAPersonalSecurity` object
java.lang.String	**toString**()

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Field Detail

preSharedKey

`public java.lang.String` **`preSharedKey`**

The 256-bit pre-shared key (PSK) formatted as a hexidecimal string.

The string shall consist of 64 hexadecimal digits.

---

cipherType

`public short` **`cipherType`**

Defines the encryption mechanism to be used.

See Also:
`NMS.CIPHER_CCMP`, `NMS.CIPHER_TKIP`

Constructor Detail

NMS.WPAPersonalSecurity

`public` **`NMS.WPAPersonalSecurity`**`()`

Default constructor

Method Detail

toString

`public java.lang.String` **`toString`**`()`

Overrides:
`toString` in class `java.lang.Object`

---

toObjectNotation

`public java.lang.String` **`toObjectNotation`**`()`

Returns a string containing the object notation representation of the `WPAPersonalSecurity` object

Returns:
the object notation string

Package com.meshdynamics.api

Interface Summary	
NMS.ConnectedDevice	Defines the properties of all devices connected to a `NMS.Node`
NMS.NeighborNode	Defines the properties of all neighbor nodes detected by a `NMS.Node`
NMS.Network	The `Network` interface defines all properties and actions associated with a mesh network.
NMS.NetworkListener	The `NetworkListener` interface is used to receive events on a mesh network.
NMS.Node	The `Node` interface defines all the properties and actions that can be carried out on a mesh node.
NMS.Thread.Runnable	The `Runnable` interface is implemented by any class whose instances are executed by a thread.

Class Summary	
NMS	NMS is the primary class for using the **Meshdynamics Network Management System (NMS) API**.
NMS.ACLConfiguration	Defines the Access Control List configuration for a node.
NMS.ACLEntry	Defines an Access Control List entry.
NMS.EffistreamRule	Defines a Effistream QoS rule.
NMS.GeneralConfiguration	Defines all Node level fields used by a `NMS.Node`.
NMS.Hashtable	The Hashtable class provides an implementation of a Hashtable of generic 'Object' keys and generic 'Object' values.
NMS.InterfaceConfiguration	Defines the interface level settings for a `NMS.Node`.
NMS.ObjectArray	The ObjectArray class provides an interface to a growable array that stores object references.
NMS.ShortArray	Defines an array of short integers.
NMS.Thread	The `Thread` class provides multi-threading functionality to scripting platforms.
NMS.VlanConfiguration	Defines the settings for a Virtual-LAN in a `NMS.Node`.
NMS.WEPSecurity	Defines the information used by the IEEE 802.11 **Wired Equivalent Privacy** (WEP) setting by a Node's downlink interface.
NMS.WPAEnterpriseSecurity	Defines the information used for the Wifi Protected Access security setting by a Node's downlink interface in an enterprise environment.
NMS.WPAPersonalSecurity	Defines the information used for the Wifi Protected Access (WPA) security setting by a node's downlink interface.

Hierarchy For Package com.meshdynamics.api

Class Hierarchy

- java.lang.Object
  - com.meshdynamics.api.**NMS**
  - com.meshdynamics.api.**NMS.ACLConfiguration**
  - com.meshdynamics.api.**NMS.ACLEntry**
  - com.meshdynamics.api.**NMS.EffistreamRule**
  - com.meshdynamics.api.**NMS.GeneralConfiguration**
  - com.meshdynamics.api.**NMS.Hashtable**
  - com.meshdynamics.api.**NMS.InterfaceConfiguration**
  - com.meshdynamics.api.**NMS.ObjectArray**
  - com.meshdynamics.api.**NMS.ShortArray**
  - com.meshdynamics.api.**NMS.VlanConfiguration**
  - com.meshdynamics.api.**NMS.WEPSecurity**
  - com.meshdynamics.api.**NMS.WPAEnterpriseSecurity**
  - com.meshdynamics.api.**NMS.WPAPersonalSecurity**
  - java.lang.Thread (implements java.lang.Runnable)
    - com.meshdynamics.api.**NMS.Thread**

Interface Hierarchy

- com.meshdynamics.api.**NMS.ConnectedDevice**
- com.meshdynamics.api.**NMS.NeighborNode**
- com.meshdynamics.api.**NMS.Network**
- com.meshdynamics.api.**NMS.NetworkListener**
- com.meshdynamics.api.**NMS.Node**
- com.meshdynamics.api.**NMS.Thread.Runnable**

Deprecated API

Contents

Package Class Tree Deprecated Index Help
PREV NEXT FRAMES NO FRAMES All Classes

A

ackTimeout - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The timeout in multiples of `10` micro-seconds for the 802.11 ACK frame.
actionBitRate - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Specifies that the transmit rate.
This field is only valid for leaf-level rules.
actionDot11eCategory - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Specifies that the IEEE 802.11e category.
actionDropPacket - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Specifies that the packets will be dropped.
actionNoAck - Variable in class com.meshdynamics.api.NMS.EffistreamRule
When non-zero specifies that the packets will be sent without acknowledgement.
actionQueuedRetry - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Specifies that the transmit rate.
add(Object) - Method in class com.meshdynamics.api.NMS.ObjectArray
Add a object reference to the end of the array and increase the length by 1.
addChild(NMS.EffistreamRule) - Method in class com.meshdynamics.api.NMS.EffistreamRule
Adds a child rule to the rule object.
addEntry(NMS.ACLEntry) - Method in class com.meshdynamics.api.NMS.ACLConfiguration
Adds the entry into the entries array.
addListener(NMS.NetworkListener) - Method in interface com.meshdynamics.api.NMS.Network
Adds the specified `NetworkListener` callback hook to the mesh network.
addVlan(NMS.VlanConfiguration) - Method in interface com.meshdynamics.api.NMS.Node
Adds the specified VLAN to the `Node`.
allowClientConnection - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
When set to `0`, does not allow standard 802.11 clients to connect to the interface, and only allows child mesh nodes to connect to the interface.

B

beginConfigurationUpdate() - Method in interface com.meshdynamics.api.NMS.Node
Starts a configuration transaction bracket.
block - Variable in class com.meshdynamics.api.NMS.ACLEntry
Set to non-zero to block the device.
bytesToHexString(byte[]) - Static method in class com.meshdynamics.api.NMS
This utility method converts a byte array to a hexadecimal string.

C

cancelConfigurationUpdate() - Method in interface com.meshdynamics.api.NMS.Node
Closes the current configuration transaction bracket without sending the configuration update.
CIPHER_CCMP - Static variable in class com.meshdynamics.api.NMS
Specifies AES-CCMP encryption to be used for WPA/WPA2 Personal/Enterprise security options.

CIPHER_TKIP - Static variable in class com.meshdynamics.api.NMS
Specifies TKIP encryption to be used for WPA/WPA2 Personal/Enterprise security options.
cipherType - Variable in class com.meshdynamics.api.NMS.WPAEnterpriseSecurity
Defines the encryption mechanism to be used.
cipherType - Variable in class com.meshdynamics.api.NMS.WPAPersonalSecurity
Defines the encryption mechanism to be used.
clear() - Method in class com.meshdynamics.api.NMS.Hashtable
Clears the hashtable.
clear() - Method in class com.meshdynamics.api.NMS.ObjectArray
Removes all elements in the array and sets the number of elements to 0.
closeNetwork(NMS.Network) - Method in class com.meshdynamics.api.NMS
Closes the specified network.
com.meshdynamics.api - package com.meshdynamics.api

commitConfigurationUpdate() - Method in interface com.meshdynamics.api.NMS.Node
Closes the current configuration transaction bracket and sends the updated configuration to the `Node`.
COUNTRY_CODE_CUSTOM - Static variable in class com.meshdynamics.api.NMS
Specifies the use of custom channels.
COUNTRY_CODE_DEFAULT - Static variable in class com.meshdynamics.api.NMS
Specifies the default country code for node operation.
countryCode - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The operating country code for the node.

D

dcaList - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
When `dynamicChannelAllocation` is non-zero, downlink interfaces choose the best channel from the integers specified in this array.
deleteNode(NMS.Node) - Method in interface com.meshdynamics.api.NMS.Network
Deletes the specified node from the mesh network.
dfsRequired - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
Specifies whether Dynamics Frequency Selection and RADAR detection is required for the `regulatoryDomain`.
dot11eCategory - Variable in class com.meshdynamics.api.NMS.ACLEntry
The IEEE 802.11e access category for the device.
dot11eCategory - Variable in class com.meshdynamics.api.NMS.VlanConfiguration
The IEEE 802.11e access category to be used for packets for the VLAN.
dot11eEnabled - Variable in class com.meshdynamics.api.NMS.ACLEntry
Set to non-zero of `dot11eCategory` is valid.
dot11eEnabled - Variable in class com.meshdynamics.api.NMS.VlanConfiguration
Non-zero if IEEE 802.11e based QOS is enabled for the VLAN.
dot1pPriority - Variable in class com.meshdynamics.api.NMS.VlanConfiguration
The IEEE 802.1p bridge priority for the VLAN.
dynamicChannelAllocation - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The node level flag determining whether downlink interfaces shall use the Dynamic Channel Allocation scheme.
dynamicChannelAllocation - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
When set to `0`, disables dynamic channel allocation and forces the interface to use the channel specified by `manualChannel`.

E

EFFISTREAM_MATCH_ETH_DST - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the ETHERNET destination address field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a MAC-address.
EFFISTREAM_MATCH_ETH_SRC - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the ETHERNET source address field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a MAC-address.
EFFISTREAM_MATCH_ETH_TYPE - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the ETHERNET type field.
EFFISTREAM_MATCH_IGNORE - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code used at the ROOT level.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.
EFFISTREAM_MATCH_IP_DIFFSRV - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the IP Diffrentiated services field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.
EFFISTREAM_MATCH_IP_DST - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the IP destination address field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a IP-address.
EFFISTREAM_MATCH_IP_PROTO - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the IP protocol field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.
EFFISTREAM_MATCH_IP_SRC - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the IP source address field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a IP-address.
EFFISTREAM_MATCH_IP_TOS - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the IP Type-of-Service field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.
EFFISTREAM_MATCH_RTP_LENGTH - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the RTP data length.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a :).
EFFISTREAM_MATCH_RTP_PAYLOAD - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the RTP payload code field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.
EFFISTREAM_MATCH_RTP_VERSION - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the RTP version field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.
EFFISTREAM_MATCH_TCP_DST_PORT - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the TCP destination port field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a :).
EFFISTREAM_MATCH_TCP_LENGTH - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the TCP segment length.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a :).
EFFISTREAM_MATCH_TCP_SRC_PORT - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the TCP source port field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a :).
EFFISTREAM_MATCH_UDP_DST_PORT - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the UDP destination port field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a :).
EFFISTREAM_MATCH_UDP_LENGTH - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the UDP datagram length.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a :).

EFFISTREAM_MATCH_UDP_SRC_PORT - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the UDP source port field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).
entries - Variable in class com.meshdynamics.api.NMS.ACLConfiguration
The array of `NMS.ACLEntry` objects.
essid - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The ESSID used in 802.11 beacons and 802.11 probe-response packets transmitted by the downlink interface.
essid - Variable in class com.meshdynamics.api.NMS.VlanConfiguration
The ESSID used in 802.11 probe-response packets.
EVENT_NETWORK_CLOSE - Static variable in class com.meshdynamics.api.NMS
Specifies that a network was closed.
EVENT_NODE_DEAD - Static variable in class com.meshdynamics.api.NMS
Specifies that a node in unreachable in the mesh network.
EVENT_NODE_HEARTBEAT - Static variable in class com.meshdynamics.api.NMS
Specifies that a heartbeat was received from a node in the mesh network.
EVENT_NODE_HEARTBEAT_MISS - Static variable in class com.meshdynamics.api.NMS
Specifies that a node's heartbeat was missed in the mesh network.
EVENT_NODE_SCAN - Static variable in class com.meshdynamics.api.NMS
Specifies that a node is conducting dynamic channel allocation scan.
executeCommand(String) - Method in interface com.meshdynamics.api.NMS.Node
Executes a Meshdynamics MeshCommand™ on the `Node`.

F

firstChild - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Reference to the next child rule object.
fragThreshold - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The 802.11 fragmentation threshold for the interface.
fromXmlSpec(String) - Static method in class com.meshdynamics.api.NMS.EffistreamRule
Returns a EffistreamRule object hierarchy based on a XML based input.

G

gatewayIpAddress - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The ip-address of the default gateway in dotted decimal form.
generateConfigMacro(String) - Method in interface com.meshdynamics.api.NMS.Node
Generates a configuration macro script for the `Node`.
get(Object) - Method in class com.meshdynamics.api.NMS.Hashtable
Retrieves the value for the specified key.
get(int) - Method in class com.meshdynamics.api.NMS.ObjectArray
Retrieves the object reference at the specified index.
get(int) - Method in class com.meshdynamics.api.NMS.ShortArray
Retrieve the value at the specified index.
getACLConfiguration() - Method in interface com.meshdynamics.api.NMS.Node
Returns the Access Control List configuration for the `Node`.
getConnectedDevices() - Method in interface com.meshdynamics.api.NMS.Node
Returns an `Enumeration` of devices that are connected to this `Node`.
getCpuUsage() - Method in interface com.meshdynamics.api.NMS.Node
Returns the current average CPU usage for the node.
getDownlinkCount() - Method in interface com.meshdynamics.api.NMS.NeighborNode

Returns the number of downlink radios seen by the node.
getEffistreamRules() - Method in interface com.meshdynamics.api.NMS.Node
Returns the Effistream™ rule hierarchy for the `Node`.
getFirmwareVersionMajor() - Method in interface com.meshdynamics.api.NMS.Node
Returns the major firmware version for the `Node`.
getFirmwareVersionMinor() - Method in interface com.meshdynamics.api.NMS.Node
Returns the minor firmware version for the `Node`.
getFirmwareVersionVariant() - Method in interface com.meshdynamics.api.NMS.Node
Returns the firmware version variant for the `Node`.
getFreeRAM() - Method in interface com.meshdynamics.api.NMS.Node
Returns the amount of free RAM in Mega-bytes.
getGeneralConfiguration() - Method in interface com.meshdynamics.api.NMS.Node
Returns the node level configuration of the `Node`.
getGpsAltitude() - Method in interface com.meshdynamics.api.NMS.Node
Returns the current operational altitude in meters.
getGpsCurrentLatitude() - Method in interface com.meshdynamics.api.NMS.Node
Returns the current operational latitude coordinate in decimal format.
getGpsCurrentLongitude() - Method in interface com.meshdynamics.api.NMS.Node
Returns the current operational longitude coordinate in decimal format.
getGpsSpeed() - Method in interface com.meshdynamics.api.NMS.Node
Returns the current operational speed in Km/Hr.
getHeartbeatSqnr() - Method in interface com.meshdynamics.api.NMS.Node
Returns the sequence number of the last heartbeat received from the node.
getHopCount() - Method in interface com.meshdynamics.api.NMS.Node
Returns the current hop level for the node.
getInputVoltage() - Method in interface com.meshdynamics.api.NMS.Node
Returns the current input voltage to the node.
getInstance() - Static method in class com.meshdynamics.api.NMS
Returns a reference to the singleton instance of the `NMS` class.
getInterfaceConfigurationByName(String) - Method in interface com.meshdynamics.api.NMS.Node
Returns the configuration of the specified interface.
getInterfaces() - Method in interface com.meshdynamics.api.NMS.Node
Returns an `Enumeration` of all interfaces in the `Node`.
getMacAddress() - Method in interface com.meshdynamics.api.NMS.ConnectedDevice
Returns the MAC address of the device formatted as a string.
getName() - Method in interface com.meshdynamics.api.NMS.Network
Returns the name of the mesh network.
getNeighborNodes() - Method in interface com.meshdynamics.api.NMS.Node
Returns an `Enumeration` of nodes that this `Node` sees as neighbors.
getNetworkByName(String) - Method in class com.meshdynamics.api.NMS
Returns a reference to a `Network` object with the specified identifier.
getNode() - Method in interface com.meshdynamics.api.NMS.NeighborNode
Returns a reference to the `NMS.Node` object representing the neighbor.
getNodeByMacAddress(String) - Method in interface com.meshdynamics.api.NMS.Network
Returns the `Node` object representing the specified MAC-address.
getNodes() - Method in interface com.meshdynamics.api.NMS.Network
Returns an `Enumeration` of all mesh nodes in the network.
getOpenNetworks() - Method in class com.meshdynamics.api.NMS
Returns an `Enumeration` of all open `Network` objects.
getParentBssid() - Method in interface com.meshdynamics.api.NMS.Node
Returns the MAC-address of the parent's downlink on which this `Node` is connected.
getParentDownlinkSignal() - Method in interface com.meshdynamics.api.NMS.Node
Returns the signal RSSI in packets received by the parent's downlink interface from this node's uplink.

getParentDownlinkTxBitRate() - Method in interface com.meshdynamics.api.NMS.Node
Returns the transmit rate used by the parent for packet's transmitted to this `Node`.
getRxSignal() - Method in interface com.meshdynamics.api.NMS.ConnectedDevice
Returns the RSSI of the packets from the device to the node.
getTemperature() - Method in interface com.meshdynamics.api.NMS.Node
Returns the current node enclosure temperature.
getTreeLinkRate() - Method in interface com.meshdynamics.api.NMS.Node
Returns the 'Tree Link Rate' for the node.
getTxBitRate() - Method in interface com.meshdynamics.api.NMS.ConnectedDevice
Returns the transmit rate of packets from the node to the device.
getUnitMacAddress() - Method in interface com.meshdynamics.api.NMS.Node
Returns the MAC address of the node formatted as a string.
getUplinkSignal() - Method in interface com.meshdynamics.api.NMS.NeighborNode
Returns the RSSI of the neighbor's first downlink signal as seen by the uplink.
getUplinkSignal(int) - Method in interface com.meshdynamics.api.NMS.NeighborNode
Returns the RSSI as seen by the uplink from the specific downlink of the neighbor.
getUplinkTxBitRate() - Method in interface com.meshdynamics.api.NMS.NeighborNode
Returns the transmit rate from the uplink to the neighbor's first downlink.
getUplinkTxBitRate(int) - Method in interface com.meshdynamics.api.NMS.NeighborNode
Returns the transmit rate from the uplink to the specfic downlink of the neighbor.
getVlanConfigurationByTag(short) - Method in interface com.meshdynamics.api.NMS.Node
Returns the configuration of the specified VLAN.
getVlans() - Method in interface com.meshdynamics.api.NMS.Node
Returns an `Enumeration` of all VLANS in the `Node`.
gpsLatitude - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
Latitude coordinate of the node in decimal format.
gpsLongitude - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
Longitude coordinate of the node in decimal format.

H

heartbeatInterval - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The heartbeat interval for the node.
hexStringToBytes(String) - Static method in class com.meshdynamics.api.NMS
This utility method converts a hexadecimal string into a byte array.
hideEssid - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
When non-zero causes the ESSID field of 802.11 beacons and broadcast probe-responses to contain an empty string.
hostName - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The network host-name for the node.

I

identifier - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The identifier for the interface.
INVALID_VLAN - Static variable in class com.meshdynamics.api.NMS.ACLEntry
Constant specifying the default VLAN.
ipAddress - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The ip-address for the node in dotted decimal form.
ipAddressBytesToString(byte[]) - Static method in class com.meshdynamics.api.NMS

This utlity method converts a byte representation of IP-address to a dotted decimal dormat string.
ipAddressStringToBytes(String) - Static method in class com.meshdynamics.api.NMS
This utility method converts a dotted-decimal format string IP-address to an array of bytes.
isIpReachable() - Method in interface com.meshdynamics.api.NMS.Node
Returns non-zero if this `Node` can be communicated with using IP.
isMobile() - Method in interface com.meshdynamics.api.NMS.Node
Returns whether the node is mobile or stationary.
isRemote() - Method in interface com.meshdynamics.api.NMS.Node
Returns whether the remote or local.

K

keyIndex - Variable in class com.meshdynamics.api.NMS.WEPSecurity
The index of the key used for transmitting packets.
keys() - Method in class com.meshdynamics.api.NMS.Hashtable
Returns an `Enumeration` of all the keys in the hashtable.

L

length() - Method in class com.meshdynamics.api.NMS.ObjectArray
Retrieve the number of elements in the `ObjectArray`.
length() - Method in class com.meshdynamics.api.NMS.ShortArray
Retrieve the number of elements in the `ShortArray`.

M

macAddress - Variable in class com.meshdynamics.api.NMS.ACLEntry
The MAC-address of the device.
macAddress - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The MAC address of the interface.
macAddressBytesToHexString(byte[]) - Static method in class com.meshdynamics.api.NMS
This utility method converts a byte representation of MAC-address to a string where the individual bytes are seperated by a ':' character.
macAddressHexStringToBytes(String) - Static method in class com.meshdynamics.api.NMS
This utility method converts a string repsentation of MAC-address to an array of bytes.
manualChannel - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The channel to be used when `dynamicChannelAllocation` is set to 0.
matchCriteria - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Specifies the match criteria for the rule.
matchId - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Specifies the match identifier for the rule.
maxTransmitRate - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The maximum transmit rate for the interface.
MG_CLIENT_MODE_FORWARDER - Static variable in class com.meshdynamics.api.NMS
Specifies that the Meshdynamics Management Gateway client operates as a packet forwader, forwarding all management packets from the `Node`'s to the server.
MG_CLIENT_MODE_REMOTE_MANAGER - Static variable in class com.meshdynamics.api.NMS
Specifies that the Meshdynamics Management Gateway client operates as a remote manager, receiving management packets from remote sites.

mobilityMode - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The node's mobilty mode.
model - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The model identifier for the node.

N

name - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The name of the interface.
name - Variable in class com.meshdynamics.api.NMS.VlanConfiguration
The friendly name for the VLAN.
NETWORK_TYPE_FIPS_140_2 - Static variable in class com.meshdynamics.api.NMS
Specifies that the mesh network is a FIPS 140-2 secure network.
NETWORK_TYPE_REGULAR - Static variable in class com.meshdynamics.api.NMS
Specifies that the mesh network is a regular network.
nextSibling - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Reference to the next sibling rule object.
NMS - Class in com.meshdynamics.api
NMS is the primary class for using the **Meshdynamics Network Management System (NMS) API**.
NMS() - Constructor for class com.meshdynamics.api.NMS
Protected default constructor to be used by derived classes.
NMS.ACLConfiguration - Class in com.meshdynamics.api
Defines the Access Control List configuration for a node.
NMS.ACLConfiguration() - Constructor for class com.meshdynamics.api.NMS.ACLConfiguration
Default constructor, initializes the object with an empty entries array and sets `whiteList` to `0`.
NMS.ACLConfiguration(String) - Constructor for class com.meshdynamics.api.NMS.ACLConfiguration
Constructs the `ACLConfiguration` from a object notation string.
NMS.ACLEntry - Class in com.meshdynamics.api
Defines an Access Control List entry.
NMS.ACLEntry() - Constructor for class com.meshdynamics.api.NMS.ACLEntry
Default constructor.
NMS.ConnectedDevice - Interface in com.meshdynamics.api
Defines the properties of all devices connected to a `NMS.Node`
NMS.EffistreamRule - Class in com.meshdynamics.api
Defines a Effistream QoS rule.
NMS.EffistreamRule() - Constructor for class com.meshdynamics.api.NMS.EffistreamRule
Default constructor typically used to create the 'ROOT' object for the rules.
NMS.EffistreamRule(short, String) - Constructor for class com.meshdynamics.api.NMS.EffistreamRule
Use this constructor to create a rule without specifying child rules.
NMS.EffistreamRule(short, String, NMS.EffistreamRule) - Constructor for class com.meshdynamics.api.NMS.EffistreamRule
Use this constructor to create a rule directly specifying the first child.
NMS.EffistreamRule(short, String, short, short, short, short, short) - Constructor for class com.meshdynamics.api.NMS.EffistreamRule
Use this constructor to create a leaf-level rule object.
NMS.GeneralConfiguration - Class in com.meshdynamics.api
Defines all Node level fields used by a `NMS.Node`.
NMS.GeneralConfiguration() - Constructor for class com.meshdynamics.api.NMS.GeneralConfiguration

NMS.Hashtable - Class in com.meshdynamics.api
The Hashtable class provides an implementation of a Hashtable of generic 'Object' keys and generic 'Object' values.

NMS.Hashtable() - Constructor for class com.meshdynamics.api.NMS.Hashtable
Default constructor.
NMS.InterfaceConfiguration - Class in com.meshdynamics.api
Defines the interface level settings for a `NMS.Node`.
NMS.InterfaceConfiguration() - Constructor for class com.meshdynamics.api.NMS.InterfaceConfiguration
Default constructor.
NMS.InterfaceConfiguration(String) - Constructor for class com.meshdynamics.api.NMS.InterfaceConfiguration
Initializes the configuration from the object notation string.
NMS.NeighborNode - Interface in com.meshdynamics.api
Defines the properties of all neighbor nodes detected by a `NMS.Node`
NMS.Network - Interface in com.meshdynamics.api
The `Network` interface defines all properties and actions associated with a mesh network.
NMS.NetworkListener - Interface in com.meshdynamics.api
The `NetworkListener` interface is used to receive events on a mesh network.
NMS.Node - Interface in com.meshdynamics.api
The `Node` interface defines all the properties and actions that can be carried out on a mesh node.
NMS.ObjectArray - Class in com.meshdynamics.api
The ObjectArray class provides an interface to a growable array that stores object references.
NMS.ObjectArray() - Constructor for class com.meshdynamics.api.NMS.ObjectArray
Default constructor to create the array with 0 elements.
NMS.ObjectArray(int) - Constructor for class com.meshdynamics.api.NMS.ObjectArray
Constructor to create the array with specified number of elements initialized to null.
NMS.ShortArray - Class in com.meshdynamics.api
Defines an array of short integers.
NMS.ShortArray(int) - Constructor for class com.meshdynamics.api.NMS.ShortArray
Constructs `ShortArray` object with specified number of elements.
NMS.ShortArray(short...) - Constructor for class com.meshdynamics.api.NMS.ShortArray
Constructs `ShortArray` object with the specified elements.
NMS.ShortArray(String) - Constructor for class com.meshdynamics.api.NMS.ShortArray
Constructs `ShortArray` object from a comma seperated list of numbers.
NMS.Thread - Class in com.meshdynamics.api
The `Thread` class provides multi-threading functionality to scripting platforms.
NMS.Thread(NMS.Thread.Runnable) - Constructor for class com.meshdynamics.api.NMS.Thread
Default constructor
NMS.Thread.Runnable - Interface in com.meshdynamics.api
The `Runnable` interface is implemented by any class whose instances are executed by a thread.
NMS.VlanConfiguration - Class in com.meshdynamics.api
Defines the settings for a Virtual-LAN in a `NMS.Node`.
NMS.VlanConfiguration() - Constructor for class com.meshdynamics.api.NMS.VlanConfiguration
Default constructor.
NMS.VlanConfiguration(String) - Constructor for class com.meshdynamics.api.NMS.VlanConfiguration
Creates a VlanConfiguration object from a object notation string.
NMS.WEPSecurity - Class in com.meshdynamics.api
Defines the information used by the IEEE 802.11 **Wired Equivalent Privacy** (WEP) setting by a Node's downlink interface.
NMS.WEPSecurity() - Constructor for class com.meshdynamics.api.NMS.WEPSecurity
Default constructor.
NMS.WPAEnterpriseSecurity - Class in com.meshdynamics.api
Defines the information used for the Wifi Protected Access security setting by a Node's downlink interface in an enterprise environment.
NMS.WPAEnterpriseSecurity() - Constructor for class com.meshdynamics.api.NMS.WPAEnterpriseSecurity
Default constructor
NMS.WPAPersonalSecurity - Class in com.meshdynamics.api
Defines the information used for the Wifi Protected Access (WPA) security setting by a node's downlink

interface.
NMS.WPAPersonalSecurity() - Constructor for class com.meshdynamics.api.NMS.WPAPersonalSecurity
Default constructor
nodeDescription - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
User-defined description for the node
nodeName - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
User-defined name of the node

O

onEvent(int, NMS.Network, NMS.Node) - Method in interface com.meshdynamics.api.NMS.NetworkListener
This method is called when an event occurs on the network.
openNetwork(String, String, int) - Method in class com.meshdynamics.api.NMS
Opens the specified mesh network.
operatingChannel - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The current operating channel for the interface.
OPTION_ADHOC - Static variable in class com.meshdynamics.api.NMS
Specifies that a `Node` has the Disjoint Adhoc feature option turned on.
OPTION_ADHOC_DHCP - Static variable in class com.meshdynamics.api.NMS
Specifies that a `Node` has the DHCP server option turned on.
OPTION_ADHOC_INFRA_BEGIN - Static variable in class com.meshdynamics.api.NMS
Specifies that a `Node` has the 'Begin in infrastructure' option turned on for the Disjoint Adhoc feature.
OPTION_ADHOC_SECTORED - Static variable in class com.meshdynamics.api.NMS
Specifies that a `Node` has the 'SECTORED arbitration' option turned on for the Disjoint Adhoc feature.
OPTION_FORCED_ROOT - Static variable in class com.meshdynamics.api.NMS
Specifies that a `Node` has the Forced Root feature option turned on.
OPTION_IGMP - Static variable in class com.meshdynamics.api.NMS
Specifies that a `Node` has the IGMP multicast optimization option turned on.
OPTION_LOCATION - Static variable in class com.meshdynamics.api.NMS
Specifies that a `Node` has the 802.11 PROBE request based location tracking turned on.
OPTION_SIP - Static variable in class com.meshdynamics.api.NMS
Specifies that a `Node` has the 'SIP PHONE SYSTEM' option turned on.
options - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The combination of run-time options enabled on the node.

P

parent - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Reference to the parent rule object.
PERFORMANCE_PROTOCOL_TCP - Static variable in class com.meshdynamics.api.NMS
Specifies usage of TCP protocol for running performance tests on a `Node`.
PERFORMANCE_PROTOCOL_UDP - Static variable in class com.meshdynamics.api.NMS
Specifies usage of UDP protocol for running performance tests on a `Node`.
PERFORMANCE_TYPE_DUAL_INDIVIDUAL - Static variable in class com.meshdynamics.api.NMS
Specifies that performance tests on a `Node` be run in the direction `Host -> Node` and then `Node -> Host`.
PERFORMANCE_TYPE_DUAL_SIMULTANEOUS - Static variable in class com.meshdynamics.api.NMS
Specifies that performance tests on a `Node` be run in the direction `Host -> Node` and `Node -> Host` simultaneously.
PERFORMANCE_TYPE_SINGLE - Static variable in class com.meshdynamics.api.NMS
Specifies that performance tests on a `Node` be run in the direction `Host -> Node`.

PHY_SUB_TYPE_802_11_A - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains information about a IEEE 802.11a interface.
PHY_SUB_TYPE_802_11_B - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains information about a IEEE 802.11b interface.
PHY_SUB_TYPE_802_11_BG - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains information about a mixed mode IEEE 802.11b/g interface.
PHY_SUB_TYPE_802_11_G - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains information about a IEEE 802.11g interface.
PHY_SUB_TYPE_802_11_PSF - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains information about a 20 MHz channel-width 4.9GHz interface.
PHY_SUB_TYPE_802_11_PSH - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains information about a 10 MHz channel-width 4.9GHz interface.
PHY_SUB_TYPE_802_11_PSQ - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains information about a 5 MHz channel-width 4.9GHz interface.
PHY_SUB_TYPE_IGNORE - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains information about an ETHERNET interface.
For interfaces with a `phyType` value of `PHY_TYPE_ETHERNET`, the `phySubType` shall be `PHY_SUB_TYPE_IGNORE`.
PHY_TYPE_802_11 - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains information about a IEEE 802.11 wireless interface.
PHY_TYPE_ETHERNET - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains information about an ETHERNET interface.
phySubType - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
Defines the physical layer sub-type used by the interface.
phyType - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
Defines the Physical layer used by the interface.
preferredParent - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The MAC address of the preferred parent's downlink radio.
preSharedKey - Variable in class com.meshdynamics.api.NMS.WPAPersonalSecurity
The 256-bit pre-shared key (PSK) formatted as a hexidecimal string.
put(Object, Object) - Method in class com.meshdynamics.api.NMS.Hashtable
Inserts the specified value for the specified key into the hashtable.

R

radiusServerIp - Variable in class com.meshdynamics.api.NMS.WPAEnterpriseSecurity
IP-address of the RADIUS server
radiusServerPort - Variable in class com.meshdynamics.api.NMS.WPAEnterpriseSecurity
The UDP port used by the RADIUS server
radiusServerSecret - Variable in class com.meshdynamics.api.NMS.WPAEnterpriseSecurity
The secret key used to authenticate RADIUS packets sent by the node
reboot() - Method in interface com.meshdynamics.api.NMS.Node
REBOOT's the `Node`.
rebootRequired() - Method in interface com.meshdynamics.api.NMS.Node
Returns non-zero if a 'REBOOT' is required for the `Node`.
REG_DOMAIN_CODE_CUSTOM - Static variable in class com.meshdynamics.api.NMS
Speciies the custom regulatory domain for node operation.
REG_DOMAIN_CODE_ETSI - Static variable in class com.meshdynamics.api.NMS
Specifies the ETSI regulatory domain for node operation.

REG_DOMAIN_CODE_FCC - Static variable in class com.meshdynamics.api.NMS
Specifies the FCC regulatory domain for node operation.
REG_DOMAIN_CODE_NONE - Static variable in class com.meshdynamics.api.NMS
Specifies a NULL regulatory domain for node operation.
regulatoryDomain - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The operating regulatory domain for the node.
remove(Object) - Method in class com.meshdynamics.api.NMS.Hashtable
Removes the specified key from the hashtable.
removeAt(int) - Method in class com.meshdynamics.api.NMS.ObjectArray
Removes the element at the specified index.
removeListener(NMS.NetworkListener) - Method in interface com.meshdynamics.api.NMS.Network
Removes the specified `NetworkListener` callback hook from the mesh network.
removeVlan(short) - Method in interface com.meshdynamics.api.NMS.Node
Removes the specified VLAN from the `Node`.
restoreDefaults() - Method in interface com.meshdynamics.api.NMS.Node
Restore's the `Node` to factory configuration.
rtsThreshold - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The 802.11 RTS threshold for the interface.
run() - Method in class com.meshdynamics.api.NMS.Thread

run() - Method in interface com.meshdynamics.api.NMS.Thread.Runnable
The `run` method implements the logic for the thread.
runPerformanceTest(int, short, short, int) - Method in interface com.meshdynamics.api.NMS.Node
Provides network performance information to the `Node`.

S

SECURITY_TYPE_NONE - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains no security parameters.

With this setting the `securityInfo` field of the `InterfaceConfiguration` is ignored and set to `null`.
SECURITY_TYPE_WEP_104 - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains security parameters for IEEE 802.11 WEP encryption using a 104-bit key.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WEPSecurity` object.
SECURITY_TYPE_WEP_40 - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains security parameters for IEEE 802.11 WEP encryption using a 40-bit key.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WEPSecurity` object.
SECURITY_TYPE_WPA2_ENTERPRISE - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access 2 encryption using a RADIUS server.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAEnterpriseSecurity` object.
SECURITY_TYPE_WPA2_PERSONAL - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access 2 encryption using a pre-shared key.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a

NMS.WPAPersonalSecurity object.
SECURITY_TYPE_WPA_ENTERPRISE - Static variable in class com.meshdynamics.api.NMS
Specifies that the InterfaceConfiguration object contains security parameters for Wifi Protected Access encryption using a RADIUS server.

With this setting the securityInfo field of the InterfaceConfiguration references a NMS.WPAEnterpriseSecurity object.
SECURITY_TYPE_WPA_PERSONAL - Static variable in class com.meshdynamics.api.NMS
Specifies that the InterfaceConfiguration object contains security parameters for Wifi Protected Access encryption using a pre-shared key.

With this setting the securityInfo field of the InterfaceConfiguration references a NMS.WPAPersonalSecurity object.
securityInfo - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
Opaque object containing the security settings for the interface.
securityInfo - Variable in class com.meshdynamics.api.NMS.VlanConfiguration
Opaque object containing the security settings for the VLAN.
securityType - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The encryption/authentication scheme used to secure connections on the interface.
securityType - Variable in class com.meshdynamics.api.NMS.VlanConfiguration
The encryption/authentication scheme used to secure connections on the VLAN.
set(int, Object) - Method in class com.meshdynamics.api.NMS.ObjectArray
Set the object reference at the specified index.
set(short...) - Method in class com.meshdynamics.api.NMS.ShortArray
Set the elements of the ShortArray to the specified variable argument list of numbers.
set(String) - Method in class com.meshdynamics.api.NMS.ShortArray
Set the elements of the ShortArray from a comma seperated list of numbers.
set(int, short) - Method in class com.meshdynamics.api.NMS.ShortArray
Set the value at specified index.
setACLConfiguration(NMS.ACLConfiguration) - Method in interface com.meshdynamics.api.NMS.Node
Sets the Node's Access Control List configuration.
setEffistreamRules(NMS.EffistreamRule) - Method in interface com.meshdynamics.api.NMS.Node
Updates the Effistream™ rule hierarchy for the Node.
setGeneralConfiguration(NMS.GeneralConfiguration) - Method in interface com.meshdynamics.api.NMS.Node
Updates the node level configuration for the Node.
setInterfaceConfiguration(NMS.InterfaceConfiguration) - Method in interface com.meshdynamics.api.NMS.Node
Updates the interface configuration for the Node.
setVlanConfiguration(NMS.VlanConfiguration) - Method in interface com.meshdynamics.api.NMS.Node
Sets the configuration of an existing VLAN in the Node.
setVlans(NMS.ObjectArray) - Method in interface com.meshdynamics.api.NMS.Node
Sets the Node's VLAN list from a ObjectArray.
sleep(long) - Static method in class com.meshdynamics.api.NMS.Thread
The sleep method blocks the calling thread for the specified number of milli-seconds.

Since it is a static method, the calling thread does not have to be an instance of the NMS.Thread class.
start() - Method in class com.meshdynamics.api.NMS
Starts the node detection and event generation processes for the NMS object.
start() - Method in class com.meshdynamics.api.NMS.Thread
Starts the thread.
startMGClient(short, String, int, boolean, String, String, boolean) - Method in class com.meshdynamics.api.NMS
Starts the Meshdynamics Management Gateway client for remote management.
stdErrPrintln(String) - Method in class com.meshdynamics.api.NMS
Prints the specified string to the error output stream.

stdOutPrintln(String) - Method in class com.meshdynamics.api.NMS
Prints the specified string to the standard output stream.
stop() - Method in class com.meshdynamics.api.NMS
Stops the node detection and event generation processes for the `NMS` object.
stopMGClient() - Method in class com.meshdynamics.api.NMS
Stops the Meshdynamics Management Gateway client for remote management.
subnetMask - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The subnet-mask for the node in dotted decimal form.

T

tag - Variable in class com.meshdynamics.api.NMS.VlanConfiguration
The IEEE 802.1q tag for the VLAN.
toObjectNotation() - Method in class com.meshdynamics.api.NMS.ACLConfiguration
Returns a string containing the object notation representation of the ACLConfiguration object.
toObjectNotation() - Method in class com.meshdynamics.api.NMS.ACLEntry
Returns a string containing the object notation representation of the `ACLEntry` object.
toObjectNotation() - Method in class com.meshdynamics.api.NMS.InterfaceConfiguration
Returns a string containing the object notation representation for the interface.
toObjectNotation() - Method in class com.meshdynamics.api.NMS.ObjectArray
Returns a string containing the object notation representation for the `ObjectArray`.
toObjectNotation() - Method in class com.meshdynamics.api.NMS.ShortArray
Returns a string containing the object notation representation for the `ShortArray`.
toObjectNotation() - Method in class com.meshdynamics.api.NMS.VlanConfiguration
Returns a string containing the object notation representation of the VlanConfiguration object.
toObjectNotation() - Method in class com.meshdynamics.api.NMS.WEPSecurity
Returns a string containing the object notation representation of the `WEPSecurity` object
toObjectNotation() - Method in class com.meshdynamics.api.NMS.WPAEnterpriseSecurity
Returns a string containing the object notation representation of the `WPAEnterpriseSecurity` object.
toObjectNotation() - Method in class com.meshdynamics.api.NMS.WPAPersonalSecurity
Returns a string containing the object notation representation of the `WPAPersonalSecurity` object
toString() - Method in class com.meshdynamics.api.NMS.ACLConfiguration

toString() - Method in class com.meshdynamics.api.NMS.ACLEntry

toString() - Method in class com.meshdynamics.api.NMS.EffistreamRule

toString() - Method in class com.meshdynamics.api.NMS.InterfaceConfiguration

toString() - Method in class com.meshdynamics.api.NMS.ObjectArray

toString() - Method in class com.meshdynamics.api.NMS.ShortArray

toString() - Method in class com.meshdynamics.api.NMS.VlanConfiguration

toString() - Method in class com.meshdynamics.api.NMS.WEPSecurity

toString() - Method in class com.meshdynamics.api.NMS.WPAEnterpriseSecurity

toString() - Method in class com.meshdynamics.api.NMS.WPAPersonalSecurity

toXmlSpec() - Method in class com.meshdynamics.api.NMS.EffistreamRule

Converts a EffistreamRule object hierarchy to a XML based string.

transmitPower - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The transmit power for the interface.

U

unInitialize() - Method in class com.meshdynamics.api.NMS
Un-initializes the `NMS` instance.

unInitializeInstance() - Static method in class com.meshdynamics.api.NMS
Un-initializes the singleton instance of the `NMS` class.

upgradeFirmware(String) - Method in interface com.meshdynamics.api.NMS.Node
Upgrades the firmware of the `Node`.

USAGE_TYPE_DOWNLINK - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains information about a DOWNLINK interface.

USAGE_TYPE_SCANNER - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains information about a SCANNER interface.

USAGE_TYPE_UPLINK - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains information about an UPLINK interface.

usageType - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
Defines the role in which the interface is used during the node's operation.

V

vlanTag - Variable in class com.meshdynamics.api.NMS.ACLEntry
The IEEE 802.1q VLAN tag to be used when the device associates.

W

waitForNodeDetect(String, long) - Method in interface com.meshdynamics.api.NMS.Network
Blocks the calling thread until all the nodes specified in `macAddresses` parameter are fully detected and configurable.

wepKeys - Variable in class com.meshdynamics.api.NMS.WEPSecurity
An array of upto 4 WEP keys formatted as hexadecimal strings.

whiteList - Variable in class com.meshdynamics.api.NMS.ACLConfiguration
Defines whether the ACL configuration entries specify a 'white-list'.

How This API Document Is Organized

This API (Application Programming Interface) document has pages corresponding to the items in the navigation bar, described as follows.

Package

Each package has a page that contains a list of its classes and interfaces, with a summary for each. This page can contain four categories:

- Interfaces (italic)
- Classes
- Enums
- Exceptions
- Errors
- Annotation Types

Class/Interface

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three sections consisting of a class/interface description, summary tables, and detailed member descriptions:

- Class inheritance diagram
- Direct Subclasses
- All Known Subinterfaces
- All Known Implementing Classes
- Class/interface declaration
- Class/interface description

- Nested Class Summary
- Field Summary
- Constructor Summary
- Method Summary

- Field Detail
- Constructor Detail
- Method Detail

Each summary entry contains the first sentence from the detailed description for that item. The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

Annotation Type

Each annotation type has its own separate page with the following sections:

- Annotation Type declaration
- Annotation Type description

- Required Element Summary
- Optional Element Summary
- Element Detail

Enum

Each enum has its own separate page with the following sections:

- Enum declaration
- Enum description
- Enum Constant Summary
- Enum Constant Detail

Tree (Class Hierarchy)

There is a Class Hierarchy page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. The classes are organized by inheritance structure starting with `java.lang.Object`. The interfaces do not inherit from `java.lang.Object`.

- When viewing the Overview page, clicking on "Tree" displays the hierarchy for all packages.
- When viewing a particular package, class or interface page, clicking "Tree" displays the hierarchy for only that package.

Deprecated API

The Deprecated API page lists all of the API that have been deprecated. A deprecated API is not recommended for use, generally due to improvements, and a replacement API is usually given. Deprecated APIs may be removed in future implementations.

Index

The Index contains an alphabetic list of all classes, interfaces, constructors, methods, and fields.

Prev/Next

These links take you to the next or previous class, interface, package, or related page.

Frames/No Frames

These links show and hide the HTML frames. All pages are available with or without frames.

Serialized Form

Each serializable or externalizable class has a description of its serialization fields and methods. This information is of interest to re-implementors, not to developers using the API. While there is no link in the navigation bar, you can get to this information by going to any serialized class and clicking "Serialized Form" in the "See also" section of the class description.

Constant Field Values

The Constant Field Values page lists the static final fields and their values.

This help file applies to API documentation generated using the standard doclet.

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package Class **Tree** **Deprecated** **Index** **Help**
PREV PACKAGE NEXT PACKAGE **FRAMES** **NO FRAMES**

Package com.meshdynamics.api

Interface Summary

NMS.ConnectedDevice	Defines the properties of all devices connected to a `NMS.Node`
NMS.NeighborNode	Defines the properties of all neighbor nodes detected by a `NMS.Node`
NMS.Network	The `Network` interface defines all properties and actions associated with a mesh network.
NMS.NetworkListener	The `NetworkListener` interface is used to receive events on a mesh network.
NMS.Node	The `Node` interface defines all the properties and actions that can be carried out on a mesh node.
NMS.Thread.Runnable	The `Runnable` interface is implemented by any class whose instances are executed by a thread.

Class Summary

NMS	NMS is the primary class for using the **Meshdynamics Network Management System (NMS) API**.
NMS.ACLConfiguration	Defines the Access Control List configuration for a node.
NMS.ACLEntry	Defines an Access Control List entry.
NMS.EffistreamRule	Defines a Effistream QoS rule.
NMS.GeneralConfiguration	Defines all Node level fields used by a `NMS.Node`.
NMS.Hashtable	The Hashtable class provides an implementation of a Hashtable of generic 'Object' keys and generic 'Object' values.
NMS.InterfaceConfiguration	Defines the interface level settings for a `NMS.Node`.
NMS.ObjectArray	The ObjectArray class provides an interface to a growable array that stores object references.
NMS.ShortArray	Defines an array of short integers.
NMS.Thread	The `Thread` class provides multi-threading functionality to scripting platforms.
NMS.VlanConfiguration	Defines the settings for a Virtual-LAN in a `NMS.Node`.
NMS.WEPSecurity	Defines the information used by the IEEE 802.11 **Wired Equivalent Privacy** (WEP) setting by a Node's downlink interface.
NMS.WPAEnterpriseSecurity	Defines the information used for the Wifi Protected Access security setting by a Node's downlink interface in an

	enterprise environment.
NMS.WPAPersonalSecurity	Defines the information used for the Wifi Protected Access (WPA) security setting by a node's downlink interface.

Package Class **Tree** **Deprecated** **Index** **Help**

PREV PACKAGE NEXT PACKAGE **FRAMES** **NO FRAMES**

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfiguration
NMS.Hashtable
NMS.InterfaceConfiguration
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSecurit
NMS.WPAPersonalSecurity

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES**
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Class NMS

```
java.lang.Object
  └─com.meshdynamics.api.NMS
```

```
public abstract class NMS
extends java.lang.Object
```

NMS is the primary class for using the **Meshdynamics Network Management System (NMS) API**.

It is a singleton class defining classes, interfaces and constants to be used for accessing the NMS information

All clients of the NMS API need to obtain a reference to the singleton instance of the NMS object by calling the `NMS.getInstance()` method.

Nested Class Summary

static class	**NMS.ACLConfiguration** Defines the Access Control List configuration for a node.
static class	**NMS.ACLEntry** Defines an Access Control List entry.
static interface	**NMS.ConnectedDevice** Defines the properties of all devices connected to a NMS.Node
static class	**NMS.EffistreamRule** Defines a Effistream QoS rule.
static class	**NMS.GeneralConfiguration** Defines all Node level fields used by a NMS.Node.
static class	**NMS.Hashtable** The Hashtable class provides an implementation of a Hashtable of generic 'Object' keys and generic 'Object' values.
static class	**NMS.InterfaceConfiguration** Defines the interface level settings for a NMS.Node.
static interface	**NMS.NeighborNode** Defines the properties of all neighbor nodes detected by a NMS.Node
static interface	**NMS.Network** The Network interface defines all properties and actions associated with a mesh network.
static interface	**NMS.NetworkListener** The NetworkListener interface is used to receive events on a mesh network.
static interface	**NMS.Node** The Node interface defines all the properties and actions that can be carried out on a mesh node.
static class	**NMS.ObjectArray** The ObjectArray class provides an interface to a growable array that stores object references.
static class	**NMS.ShortArray** Defines an array of short integers.
static class	**NMS.Thread** The Thread class provides multi-threading functionality to scripting platforms.

static class	**NMS.VlanConfiguration** Defines the settings for a Virtual-LAN in a NMS.Node.
static class	**NMS.WEPSecurity** Defines the information used by the IEEE 802.11 **Wired Equivalent Privacy** (WEP) setting by a Node's downlink interface.
static class	**NMS.WPAEnterpriseSecurity** Defines the information used for the Wifi Protected Access security setting by a Node's downlink interface in an enterprise environment.
static class	**NMS.WPAPersonalSecurity** Defines the information used for the Wifi Protected Access (WPA) security setting by a node's downlink interface.

Field Summary

static short	**CIPHER_CCMP** Specifies AES-CCMP encryption to be used for WPA/WPA2 Personal/Enterprise security options.
static short	**CIPHER_TKIP** Specifies TKIP encryption to be used for WPA/WPA2 Personal/Enterprise security options.
static short	**COUNTRY_CODE_CUSTOM** Specifies the use of custom channels.
static short	**COUNTRY_CODE_DEFAULT** Specifies the default country code for node operation.
static short	**EFFISTREAM_MATCH_ETH_DST** Specifies a Effistream™ match code for the ETHERNET destination address field. The matchCriteria of the EffistreamRule specifies a string containing a MAC-address.
static short	**EFFISTREAM_MATCH_ETH_SRC** Specifies a Effistream™ match code for the ETHERNET source address field. The matchCriteria of the EffistreamRule specifies a string containing a MAC-address.
static short	**EFFISTREAM_MATCH_ETH_TYPE** Specifies a Effistream™ match code for the ETHERNET type field.
static short	**EFFISTREAM_MATCH_IGNORE** Specifies a Effistream™ match code used at the ROOT level. The matchCriteria of the EffistreamRule specifies a string containing an integer.
static short	**EFFISTREAM_MATCH_IP_DIFFSRV** Specifies a Effistream™ match code for the IP Diffrentiated services field. The matchCriteria of the EffistreamRule specifies a string containing an integer.
static short	**EFFISTREAM_MATCH_IP_DST** Specifies a Effistream™ match code for the IP destination address field. The matchCriteria of the EffistreamRule specifies a string containing a IP-address.
static short	**EFFISTREAM_MATCH_IP_PROTO** Specifies a Effistream™ match code for the IP protocol field. The matchCriteria of the EffistreamRule specifies a string containing an integer.
static short	**EFFISTREAM_MATCH_IP_SRC** Specifies a Effistream™ match code for the IP source address field. The matchCriteria of the EffistreamRule specifies a string containing a IP-address.
static short	**EFFISTREAM_MATCH_IP_TOS** Specifies a Effistream™ match code for the IP Type-of-Service field. The matchCriteria of the EffistreamRule specifies a string containing an integer.
static short	**EFFISTREAM_MATCH_RTP_LENGTH** Specifies a Effistream™ match code for the RTP data length. The matchCriteria of the EffistreamRule specifies a string containing a range (two integers seperated by a :).

static short	**EFFISTREAM_MATCH_RTP_PAYLOAD** Specifies a Effistream™ match code for the RTP payload code field. The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.
static short	**EFFISTREAM_MATCH_RTP_VERSION** Specifies a Effistream™ match code for the RTP version field. The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.
static short	**EFFISTREAM_MATCH_TCP_DST_PORT** Specifies a Effistream™ match code for the TCP destination port field. The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a :).
static short	**EFFISTREAM_MATCH_TCP_LENGTH** Specifies a Effistream™ match code for the TCP segment length. The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a :).
static short	**EFFISTREAM_MATCH_TCP_SRC_PORT** Specifies a Effistream™ match code for the TCP source port field. The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a :).
static short	**EFFISTREAM_MATCH_UDP_DST_PORT** Specifies a Effistream™ match code for the UDP destination port field. The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a :).
static short	**EFFISTREAM_MATCH_UDP_LENGTH** Specifies a Effistream™ match code for the UDP datagram length. The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a :).
static short	**EFFISTREAM_MATCH_UDP_SRC_PORT** Specifies a Effistream™ match code for the UDP source port field. The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a :).
static int	**EVENT_NETWORK_CLOSE** Specifies that a network was closed.
static int	**EVENT_NODE_DEAD** Specifies that a node in unreachable in the mesh network.
static int	**EVENT_NODE_HEARTBEAT** Specifies that a heartbeat was received from a node in the mesh network.
static int	**EVENT_NODE_HEARTBEAT_MISS** Specifies that a node's heartbeat was missed in the mesh network.
static int	**EVENT_NODE_SCAN** Specifies that a node is conducting dynamic channel allocation scan.
static short	**MG_CLIENT_MODE_FORWARDER** Specifies that the Meshdynamics Management Gateway client operates as a packet forwader, forwarding all management packets from the `Node`'s to the server.
static short	**MG_CLIENT_MODE_REMOTE_MANAGER** Specifies that the Meshdynamics Management Gateway client operates as a remote manager, receiving management packets from remote sites.
static short	**NETWORK_TYPE_FIPS_140_2** Specifies that the mesh network is a FIPS 140-2 secure network.
static short	**NETWORK_TYPE_REGULAR** Specifies that the mesh network is a regular network.
static short	**OPTION_ADHOC** Specifies that a `Node` has the Disjoint Adhoc feature option turned on.
static short	**OPTION_ADHOC_DHCP**

	Specifies that a Node has the DHCP server option turned on.
static short	**OPTION_ADHOC_INFRA_BEGIN** Specifies that a Node has the 'Begin in infrastructure' option turned on for the Disjoint Adhoc feature.
static short	**OPTION_ADHOC_SECTORED** Specifies that a Node has the 'SECTORED arbitration' option turned on for the Disjoint Adhoc feature.
static short	**OPTION_FORCED_ROOT** Specifies that a Node has the Forced Root feature option turned on.
static short	**OPTION_IGMP** Specifies that a Node has the IGMP multicast optimization option turned on.
static short	**OPTION_LOCATION** Specifies that a Node has the 802.11 PROBE request based location tracking turned on.
static short	**OPTION_SIP** Specifies that a Node has the 'SIP PHONE SYSTEM' option turned on.
static short	**PERFORMANCE_PROTOCOL_TCP** Specifies usage of TCP protocol for running performance tests on a Node.
static short	**PERFORMANCE_PROTOCOL_UDP** Specifies usage of UDP protocol for running performance tests on a Node.
static short	**PERFORMANCE_TYPE_DUAL_INDIVIDUAL** Specifies that performance tests on a Node be run in the direction Host -> Node and then Node -> Host.
static short	**PERFORMANCE_TYPE_DUAL_SIMULTANEOUS** Specifies that performance tests on a Node be run in the direction Host -> Node and Node -> Host simultaneously.
static short	**PERFORMANCE_TYPE_SINGLE** Specifies that performance tests on a Node be run in the direction Host -> Node.
static short	**PHY_SUB_TYPE_802_11_A** Specifies that the InterfaceConfiguration object contains information about a IEEE 802.11a interface.
static short	**PHY_SUB_TYPE_802_11_B** Specifies that the InterfaceConfiguration object contains information about a IEEE 802.11b interface.
static short	**PHY_SUB_TYPE_802_11_BG** Specifies that the InterfaceConfiguration object contains information about a mixed mode IEEE 802.11b/g interface.
static short	**PHY_SUB_TYPE_802_11_G** Specifies that the InterfaceConfiguration object contains information about a IEEE 802.11g interface.
static short	**PHY_SUB_TYPE_802_11_PSF** Specifies that the InterfaceConfiguration object contains information about a 20 MHz channel-width 4.9GHz interface.
static short	**PHY_SUB_TYPE_802_11_PSH** Specifies that the InterfaceConfiguration object contains information about a 10 MHz channel-width 4.9GHz interface.
static short	**PHY_SUB_TYPE_802_11_PSQ** Specifies that the InterfaceConfiguration object contains information about a 5 MHz channel-width 4.9GHz interface.
static short	**PHY_SUB_TYPE_IGNORE** Specifies that the InterfaceConfiguration object contains information about an ETHERNET interface. For interfaces with a phyType value of PHY_TYPE_ETHERNET, the phySubType shall be PHY_SUB_TYPE_IGNORE.
static short	**PHY_TYPE_802_11** Specifies that the InterfaceConfiguration object contains information about a IEEE

	802.11 wireless interface.
static short	**PHY_TYPE_ETHERNET** Specifies that the `InterfaceConfiguration` object contains information about an ETHERNET interface.
static short	**REG_DOMAIN_CODE_CUSTOM** Speciies the custom regulatory domain for node operation.
static short	**REG_DOMAIN_CODE_ETSI** Specifies the ETSI regulatory domain for node operation.
static short	**REG_DOMAIN_CODE_FCC** Specifies the FCC regulatory domain for node operation.
static short	**REG_DOMAIN_CODE_NONE** Specifies a NULL regulatory domain for node operation.
static short	**SECURITY_TYPE_NONE** Specifies that the `InterfaceConfiguration` object contains no security parameters. With this setting the `securityInfo` field of the `InterfaceConfiguration` is ignored and set to `null`.
static short	**SECURITY_TYPE_WEP_104** Specifies that the `InterfaceConfiguration` object contains security parameters for IEEE 802.11 WEP encryption using a 104-bit key. With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WEPSecurity` object.
static short	**SECURITY_TYPE_WEP_40** Specifies that the `InterfaceConfiguration` object contains security parameters for IEEE 802.11 WEP encryption using a 40-bit key. With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WEPSecurity` object.
static short	**SECURITY_TYPE_WPA_ENTERPRISE** Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access encryption using a RADIUS server. With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAEnterpriseSecurity` object.
static short	**SECURITY_TYPE_WPA_PERSONAL** Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access encryption using a pre-shared key. With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAPersonalSecurity` object.
static short	**SECURITY_TYPE_WPA2_ENTERPRISE** Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access 2 encryption using a RADIUS server. With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAEnterpriseSecurity` object.
static short	**SECURITY_TYPE_WPA2_PERSONAL** Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access 2 encryption using a pre-shared key. With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAPersonalSecurity` object.
static short	**USAGE_TYPE_DOWNLINK** Specifies that the `InterfaceConfiguration` object contains information about a DOWNLINK interface.
static short	**USAGE_TYPE_SCANNER**

	Specifies that the `InterfaceConfiguration` object contains information about a SCANNER interface.
`static short`	**`USAGE_TYPE_UPLINK`** Specifies that the `InterfaceConfiguration` object contains information about an UPLINK interface.

Constructor Summary

`protected`	**`NMS`**`()` Protected default constructor to be used by derived classes.

Method Summary

`static java.lang.String`	**`bytesToHexString`**`(byte[] bytes)` This utility method converts a byte array to a hexadecimal string.
`abstract int`	**`closeNetwork`**`(NMS.Network network)` Closes the specified network.
`static NMS`	**`getInstance`**`()` Returns a reference to the singleton instance of the `NMS` class.
`abstract NMS.Network`	**`getNetworkByName`**`(java.lang.String networkName)` Returns a reference to a `Network` object with the specified identifier.
`abstract java.util.Enumeration<NMS.Network>`	**`getOpenNetworks`**`()` Returns an `Enumeration` of all open `Network` objects.
`static byte[]`	**`hexStringToBytes`**`(java.lang.String hexString)` This utility method converts a hexadecimal string into a byte array.
`static java.lang.String`	**`ipAddressBytesToString`**`(byte[] ipAddress)` This utlity method converts a byte representation of IP-address to a dotted decimal dormat string.
`static byte[]`	**`ipAddressStringToBytes`**`(java.lang.String ipAddress)` This utility method converts a dotted-decimal format string IP-address to an array of bytes.
`static java.lang.String`	**`macAddressBytesToHexString`**`(byte[] macAddress)` This utility method converts a byte representation of MAC-address to a string where the individual bytes are seperated by a ':' character.
`static byte[]`	**`macAddressHexStringToBytes`**`(java.lang.String macAddress)` This utility method converts a string repsentation of MAC-address to an array of bytes.
`abstract NMS.Network`	**`openNetwork`**`(java.lang.String networkName, java.lang.String networkKey, int networkType)` Opens the specified mesh network.
`abstract int`	**`start`**`()` Starts the node detection and event generation processes for the `NMS` object.
`abstract int`	**`startMGClient`**`(short mode, java.lang.String server, int port, boolean useSSL, java.lang.String userName, java.lang.String password, boolean ignoreLocalPackets)` Starts the Meshdynamics Management Gateway client for remote management.
`abstract void`	**`stdErrPrintln`**`(java.lang.String str)` Prints the specified string to the error output stream.
`abstract void`	**`stdOutPrintln`**`(java.lang.String str)` Prints the specified string to the standard output stream.
`abstract int`	

	stop() Stops the node detection and event generation processes for the NMS object.
abstract int	**stopMGClient**() Stops the Meshdynamics Management Gateway client for remote management.
protected abstract void	**unInitialize**() Un-initializes the NMS instance.
static void	**unInitializeInstance**() Un-initializes the singleton instance of the NMS class.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

USAGE_TYPE_UPLINK

```
public static final short USAGE_TYPE_UPLINK
```

Specifies that the InterfaceConfiguration object contains information about an UPLINK interface.

See Also:
NMS.InterfaceConfiguration.usageType, Constant Field Values

USAGE_TYPE_DOWNLINK

```
public static final short USAGE_TYPE_DOWNLINK
```

Specifies that the InterfaceConfiguration object contains information about a DOWNLINK interface.

See Also:
NMS.InterfaceConfiguration.usageType, Constant Field Values

USAGE_TYPE_SCANNER

```
public static final short USAGE_TYPE_SCANNER
```

Specifies that the InterfaceConfiguration object contains information about a SCANNER interface.

See Also:
NMS.InterfaceConfiguration.usageType, Constant Field Values

PHY_TYPE_ETHERNET

```
public static final short PHY_TYPE_ETHERNET
```

Specifies that the InterfaceConfiguration object contains information about an ETHERNET interface.

See Also:
NMS.InterfaceConfiguration.phyType, Constant Field Values

PHY_TYPE_802_11

```
public static final short PHY_TYPE_802_11
```

Specifies that the `InterfaceConfiguration` object contains information about a IEEE 802.11 wireless interface.

See Also:
`NMS.InterfaceConfiguration.phyType`, Constant Field Values

PHY_SUB_TYPE_IGNORE

```
public static final short PHY_SUB_TYPE_IGNORE
```

Specifies that the `InterfaceConfiguration` object contains information about an ETHERNET interface. For interfaces with a `phyType` value of `PHY_TYPE_ETHERNET`, the `phySubType` shall be `PHY_SUB_TYPE_IGNORE`.

See Also:
`NMS.InterfaceConfiguration.phyType`, `NMS.InterfaceConfiguration.phySubType`, Constant Field Values

PHY_SUB_TYPE_802_11_A

```
public static final short PHY_SUB_TYPE_802_11_A
```

Specifies that the `InterfaceConfiguration` object contains information about a IEEE 802.11a interface.

See Also:
`NMS.InterfaceConfiguration.phySubType`, Constant Field Values

PHY_SUB_TYPE_802_11_B

```
public static final short PHY_SUB_TYPE_802_11_B
```

Specifies that the `InterfaceConfiguration` object contains information about a IEEE 802.11b interface.

See Also:
`NMS.InterfaceConfiguration.phySubType`, Constant Field Values

PHY_SUB_TYPE_802_11_G

```
public static final short PHY_SUB_TYPE_802_11_G
```

Specifies that the `InterfaceConfiguration` object contains information about a IEEE 802.11g interface.

See Also:
`NMS.InterfaceConfiguration.phySubType`, Constant Field Values

PHY_SUB_TYPE_802_11_BG

```
public static final short PHY_SUB_TYPE_802_11_BG
```

Specifies that the `InterfaceConfiguration` object contains information about a mixed mode IEEE 802.11b/g interface.

See Also:
`NMS.InterfaceConfiguration.phySubType`, Constant Field Values

PHY_SUB_TYPE_802_11_PSQ

```
public static final short PHY_SUB_TYPE_802_11_PSQ
```

Specifies that the `InterfaceConfiguration` object contains information about a 5 MHz channel-width 4.9GHz interface.

See Also:
`NMS.InterfaceConfiguration.phySubType`, Constant Field Values

PHY_SUB_TYPE_802_11_PSH

`public static final short` **`PHY_SUB_TYPE_802_11_PSH`**

Specifies that the `InterfaceConfiguration` object contains information about a 10 MHz channel-width 4.9GHz interface.

See Also:
`NMS.InterfaceConfiguration.phySubType`, Constant Field Values

PHY_SUB_TYPE_802_11_PSF

`public static final short` **`PHY_SUB_TYPE_802_11_PSF`**

Specifies that the `InterfaceConfiguration` object contains information about a 20 MHz channel-width 4.9GHz interface.

See Also:
`NMS.InterfaceConfiguration.phySubType`, Constant Field Values

SECURITY_TYPE_NONE

`public static final short` **`SECURITY_TYPE_NONE`**

Specifies that the `InterfaceConfiguration` object contains no security parameters.

With this setting the `securityInfo` field of the `InterfaceConfiguration` is ignored and set to `null`.

See Also:
`NMS.InterfaceConfiguration.securityType`, Constant Field Values

SECURITY_TYPE_WEP_40

`public static final short` **`SECURITY_TYPE_WEP_40`**

Specifies that the `InterfaceConfiguration` object contains security parameters for IEEE 802.11 WEP encryption using a 40-bit key.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WEPSecurity` object.

See Also:
`NMS.InterfaceConfiguration.securityType`, `NMS.WEPSecurity`, Constant Field Values

SECURITY_TYPE_WEP_104

`public static final short` **`SECURITY_TYPE_WEP_104`**

Specifies that the `InterfaceConfiguration` object contains security parameters for IEEE 802.11 WEP encryption using a 104-bit key.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WEPSecurity` object.

See Also:
NMS.InterfaceConfiguration.securityType, NMS.WEPSecurity, Constant Field Values

---

SECURITY_TYPE_WPA_PERSONAL

`public static final short` **`SECURITY_TYPE_WPA_PERSONAL`**

Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access encryption using a pre-shared key.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAPersonalSecurity` object.

See Also:
NMS.InterfaceConfiguration.securityType, NMS.WPAPersonalSecurity, Constant Field Values

---

SECURITY_TYPE_WPA_ENTERPRISE

`public static final short` **`SECURITY_TYPE_WPA_ENTERPRISE`**

Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access encryption using a RADIUS server.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAEnterpriseSecurity` object.

See Also:
NMS.InterfaceConfiguration.securityType, NMS.WPAEnterpriseSecurity, Constant Field Values

---

SECURITY_TYPE_WPA2_PERSONAL

`public static final short` **`SECURITY_TYPE_WPA2_PERSONAL`**

Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access 2 encryption using a pre-shared key.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAPersonalSecurity` object.

See Also:
NMS.InterfaceConfiguration.securityType, NMS.WPAPersonalSecurity, Constant Field Values

---

SECURITY_TYPE_WPA2_ENTERPRISE

`public static final short` **`SECURITY_TYPE_WPA2_ENTERPRISE`**

Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access 2 encryption using a RADIUS server.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAEnterpriseSecurity` object.

See Also:
NMS.InterfaceConfiguration.securityType, NMS.WPAEnterpriseSecurity, Constant Field Values

---

CIPHER_CCMP

public static final short **CIPHER_CCMP**

Specifies AES-CCMP encryption to be used for WPA/WPA2 Personal/Enterprise security options.

See Also:
NMS.InterfaceConfiguration.securityType, NMS.WPAPersonalSecurity.cipherType, NMS.WPAEnterpriseSecurity.cipherType, Constant Field Values

---

CIPHER_TKIP

public static final short **CIPHER_TKIP**

Specifies TKIP encryption to be used for WPA/WPA2 Personal/Enterprise security options.

See Also:
NMS.InterfaceConfiguration.securityType, NMS.WPAPersonalSecurity.cipherType, NMS.WPAEnterpriseSecurity.cipherType, Constant Field Values

---

EVENT_NODE_HEARTBEAT

public static final int **EVENT_NODE_HEARTBEAT**

Specifies that a heartbeat was received from a node in the mesh network.

See Also:
NMS.NetworkListener.onEvent(int, com.meshdynamics.api.NMS.Network, com.meshdynamics.api.NMS.Node), Constant Field Values

---

EVENT_NODE_HEARTBEAT_MISS

public static final int **EVENT_NODE_HEARTBEAT_MISS**

Specifies that a node's heartbeat was missed in the mesh network.

See Also:
NMS.NetworkListener.onEvent(int, com.meshdynamics.api.NMS.Network, com.meshdynamics.api.NMS.Node), Constant Field Values

---

EVENT_NODE_DEAD

public static final int **EVENT_NODE_DEAD**

Specifies that a node in unreachable in the mesh network.

See Also:
NMS.NetworkListener.onEvent(int, com.meshdynamics.api.NMS.Network, com.meshdynamics.api.NMS.Node), Constant Field Values

---

EVENT_NODE_SCAN

public static final int **EVENT_NODE_SCAN**

Specifies that a node is conducting dynamic channel allocation scan.

See Also:
NMS.NetworkListener.onEvent(int, com.meshdynamics.api.NMS.Network, com.meshdynamics.api.NMS.Node), Constant Field Values

---

EVENT_NETWORK_CLOSE

`public static final int` **`EVENT_NETWORK_CLOSE`**

Specifies that a network was closed.

See Also:
`NMS.NetworkListener.onEvent(int, com.meshdynamics.api.NMS.Network, com.meshdynamics.api.NMS.Node)`, Constant Field Values

OPTION_IGMP

`public static final short` **`OPTION_IGMP`**

Specifies that a `Node` has the IGMP multicast optimization option turned on.

See Also:
`NMS.GeneralConfiguration.options`, Constant Field Values

OPTION_ADHOC

`public static final short` **`OPTION_ADHOC`**

Specifies that a `Node` has the Disjoint Adhoc feature option turned on.

See Also:
`NMS.GeneralConfiguration.options`, Constant Field Values

OPTION_FORCED_ROOT

`public static final short` **`OPTION_FORCED_ROOT`**

Specifies that a `Node` has the Forced Root feature option turned on.

See Also:
`NMS.GeneralConfiguration.options`, Constant Field Values

OPTION_ADHOC_INFRA_BEGIN

`public static final short` **`OPTION_ADHOC_INFRA_BEGIN`**

Specifies that a `Node` has the 'Begin in infrastructure' option turned on for the Disjoint Adhoc feature.

See Also:
`NMS.GeneralConfiguration.options`, Constant Field Values

OPTION_ADHOC_DHCP

`public static final short` **`OPTION_ADHOC_DHCP`**

Specifies that a `Node` has the DHCP server option turned on.

See Also:
`NMS.GeneralConfiguration.options`, Constant Field Values

OPTION_LOCATION

`public static final short` **`OPTION_LOCATION`**

Specifies that a `Node` has the 802.11 PROBE request based location tracking turned on.

See Also:

NMS.GeneralConfiguration.options, Constant Field Values

OPTION_ADHOC_SECTORED

```
public static final short OPTION_ADHOC_SECTORED
```

Specifies that a Node has the 'SECTORED arbitration' option turned on for the Disjoint Adhoc feature.

See Also:
NMS.GeneralConfiguration.options, Constant Field Values

OPTION_SIP

```
public static final short OPTION_SIP
```

Specifies that a Node has the 'SIP PHONE SYSTEM' option turned on.

See Also:
NMS.GeneralConfiguration.options, Constant Field Values

NETWORK_TYPE_REGULAR

```
public static final short NETWORK_TYPE_REGULAR
```

Specifies that the mesh network is a regular network.

See Also:
openNetwork(java.lang.String, java.lang.String, int), Constant Field Values

---

NETWORK_TYPE_FIPS_140_2

```
public static final short NETWORK_TYPE_FIPS_140_2
```

Specifies that the mesh network is a FIPS 140-2 secure network.

See Also:
openNetwork(java.lang.String, java.lang.String, int), Constant Field Values

---

EFFISTREAM_MATCH_IGNORE

```
public static final short EFFISTREAM_MATCH_IGNORE
```

Specifies a Effistream™ match code used at the ROOT level.
The matchCriteria of the EffistreamRule specifies a string containing an integer.

See Also:
NMS.EffistreamRule.matchId, NMS.EffistreamRule.matchCriteria, Constant Field Values

---

EFFISTREAM_MATCH_ETH_TYPE

```
public static final short EFFISTREAM_MATCH_ETH_TYPE
```

Specifies a Effistream™ match code for the ETHERNET type field.

See Also:
NMS.EffistreamRule.matchId, NMS.EffistreamRule.matchCriteria, Constant Field Values

---

EFFISTREAM_MATCH_ETH_DST

`public static final short` **`EFFISTREAM_MATCH_ETH_DST`**

Specifies a Effistream™ match code for the ETHERNET destination address field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a MAC-address.

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_ETH_SRC

`public static final short` **`EFFISTREAM_MATCH_ETH_SRC`**

Specifies a Effistream™ match code for the ETHERNET source address field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a MAC-address.

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_IP_TOS

`public static final short` **`EFFISTREAM_MATCH_IP_TOS`**

Specifies a Effistream™ match code for the IP Type-of-Service field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_IP_DIFFSRV

`public static final short` **`EFFISTREAM_MATCH_IP_DIFFSRV`**

Specifies a Effistream™ match code for the IP Diffrentiated services field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_IP_SRC

`public static final short` **`EFFISTREAM_MATCH_IP_SRC`**

Specifies a Effistream™ match code for the IP source address field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a IP-address.

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_IP_DST

`public static final short` **`EFFISTREAM_MATCH_IP_DST`**

Specifies a Effistream™ match code for the IP destination address field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a IP-address.

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_IP_PROTO

`public static final short` **`EFFISTREAM_MATCH_IP_PROTO`**

Specifies a Effistream™ match code for the IP protocol field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_UDP_SRC_PORT

`public static final short` **`EFFISTREAM_MATCH_UDP_SRC_PORT`**

Specifies a Effistream™ match code for the UDP source port field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_UDP_DST_PORT

`public static final short` **`EFFISTREAM_MATCH_UDP_DST_PORT`**

Specifies a Effistream™ match code for the UDP destination port field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_UDP_LENGTH

`public static final short` **`EFFISTREAM_MATCH_UDP_LENGTH`**

Specifies a Effistream™ match code for the UDP datagram length.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_TCP_SRC_PORT

`public static final short` **`EFFISTREAM_MATCH_TCP_SRC_PORT`**

Specifies a Effistream™ match code for the TCP source port field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_TCP_DST_PORT

`public static final short` **`EFFISTREAM_MATCH_TCP_DST_PORT`**

Specifies a Effistream™ match code for the TCP destination port field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_TCP_LENGTH

`public static final short` **`EFFISTREAM_MATCH_TCP_LENGTH`**

Specifies a Effistream™ match code for the TCP segment length.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_RTP_VERSION

`public static final short` **`EFFISTREAM_MATCH_RTP_VERSION`**

Specifies a Effistream™ match code for the RTP version field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_RTP_PAYLOAD

`public static final short` **`EFFISTREAM_MATCH_RTP_PAYLOAD`**

Specifies a Effistream™ match code for the RTP payload code field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

EFFISTREAM_MATCH_RTP_LENGTH

`public static final short` **`EFFISTREAM_MATCH_RTP_LENGTH`**

Specifies a Effistream™ match code for the RTP data length.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

See Also:
`NMS.EffistreamRule.matchId`, `NMS.EffistreamRule.matchCriteria`, Constant Field Values

PERFORMANCE_PROTOCOL_TCP

`public static final short` **`PERFORMANCE_PROTOCOL_TCP`**

Specifies usage of TCP protocol for running performance tests on a `Node`.

See Also:
`NMS.Node.runPerformanceTest(int, short, short, int)`, Constant Field Values

PERFORMANCE_PROTOCOL_UDP

`public static final short` **`PERFORMANCE_PROTOCOL_UDP`**

Specifies usage of UDP protocol for running performance tests on a `Node`.

See Also:
`NMS.Node.runPerformanceTest(int, short, short, int)`, Constant Field Values

PERFORMANCE_TYPE_SINGLE

`public static final short` **`PERFORMANCE_TYPE_SINGLE`**

Specifies that performance tests on a `Node` be run in the direction `Host` -> `Node`.

See Also:
`NMS.Node.runPerformanceTest(int, short, short, int)`, Constant Field Values

PERFORMANCE_TYPE_DUAL_INDIVIDUAL

`public static final short` **`PERFORMANCE_TYPE_DUAL_INDIVIDUAL`**

Specifies that performance tests on a `Node` be run in the direction `Host` -> `Node` and then `Node` -> `Host`.

See Also:
`NMS.Node.runPerformanceTest(int, short, short, int)`, Constant Field Values

PERFORMANCE_TYPE_DUAL_SIMULTANEOUS

`public static final short` **`PERFORMANCE_TYPE_DUAL_SIMULTANEOUS`**

Specifies that performance tests on a `Node` be run in the direction `Host` -> `Node` and `Node` -> `Host` simultaneously.

See Also:
`NMS.Node.runPerformanceTest(int, short, short, int)`, Constant Field Values

MG_CLIENT_MODE_FORWARDER

`public static final short` **`MG_CLIENT_MODE_FORWARDER`**

Specifies that the Meshdynamics Management Gateway client operates as a packet forwader, forwarding all management packets from the `Node`'s to the server.

See Also:
`startMGClient(short, java.lang.String, int, boolean, java.lang.String, java.lang.String, boolean)`, Constant Field Values

MG_CLIENT_MODE_REMOTE_MANAGER

`public static final short` **`MG_CLIENT_MODE_REMOTE_MANAGER`**

Specifies that the Meshdynamics Management Gateway client operates as a remote manager, receiving management packets from remote sites.

See Also:
`startMGClient(short, java.lang.String, int, boolean, java.lang.String, java.lang.String, boolean)`, Constant Field Values

COUNTRY_CODE_DEFAULT

```
public static final short COUNTRY_CODE_DEFAULT
```

Specifies the default country code for node operation.

See Also:
Constant Field Values

COUNTRY_CODE_CUSTOM

```
public static final short COUNTRY_CODE_CUSTOM
```

Specifies the use of custom channels.

This is only allowed via the use of the Meshdynamics RF-Editor API.

See Also:
Constant Field Values

REG_DOMAIN_CODE_NONE

```
public static final short REG_DOMAIN_CODE_NONE
```

Specifies a NULL regulatory domain for node operation.

See Also:
Constant Field Values

REG_DOMAIN_CODE_FCC

```
public static final short REG_DOMAIN_CODE_FCC
```

Specifies the FCC regulatory domain for node operation.

See Also:
Constant Field Values

REG_DOMAIN_CODE_ETSI

```
public static final short REG_DOMAIN_CODE_ETSI
```

Specifies the ETSI regulatory domain for node operation.

See Also:
Constant Field Values

REG_DOMAIN_CODE_CUSTOM

```
public static final short REG_DOMAIN_CODE_CUSTOM
```

Speciies the custom regulatory domain for node operation.

This is only allowed via the use of the Meshdynamics RF-Editor API.

See Also:
Constant Field Values

Constructor Detail

NMS

```
protected NMS()
```

Protected default constructor to be used by derived classes.

Method Detail

getInstance

```
public static NMS getInstance()
```

Returns a reference to the singleton instance of the `NMS` class.

Returns:
reference to the `NMS` instance

---

unInitializeInstance

```
public static void unInitializeInstance()
```

Un-initializes the singleton instance of the `NMS` class.

---

hexStringToBytes

```
public static byte[] hexStringToBytes(java.lang.String hexString)
```

This utility method converts a hexadecimal string into a byte array.

Parameters:
`hexString` - the hexadecimal string
Returns:
byte array containing the byte representation of the hexadecimal string
See Also:
`bytesToHexString(byte[])`

---

bytesToHexString

```
public static java.lang.String bytesToHexString(byte[] bytes)
```

This utility method converts a byte array to a hexadecimal string.

Parameters:
`bytes` - the byte array to be converted.
Returns:
hexadecimal string
See Also:
`hexStringToBytes(java.lang.String)`

---

macAddressBytesToHexString

```
public static java.lang.String macAddressBytesToHexString(byte[] macAddress)
```

This utility method converts a byte representation of MAC-address to a string where the individual bytes are seperated by a ':' character.

Parameters:
`macAddress` - byte array containing the MAC address
Returns:
string representation of the MAC address

See Also:
macAddressHexStringToBytes(java.lang.String)

---

ipAddressBytesToString

```
public static java.lang.String ipAddressBytesToString(byte[] ipAddress)
```

This utlity method converts a byte representation of IP-address to a dotted decimal dormat string.

Parameters:
`ipAddress` - the byte array containing the IP-address
Returns:
dotted decimal format string representation of the IP-address
See Also:
ipAddressStringToBytes(java.lang.String)

---

macAddressHexStringToBytes

```
public static byte[] macAddressHexStringToBytes(java.lang.String macAddress)
```

This utility method converts a string repsentation of MAC-address to an array of bytes.

Parameters:
`macAddress` - the string representation of the MAC-address.
Returns:
byte array containing the MAC-address
See Also:
macAddressBytesToHexString(byte[])

---

ipAddressStringToBytes

```
public static byte[] ipAddressStringToBytes(java.lang.String ipAddress)
```

This utility method converts a dotted-decimal format string IP-address to an array of bytes.

Parameters:
`ipAddress` - the dotted-decimal string IP-address.
Returns:
byte array containing the IP-address
See Also:
ipAddressBytesToString(byte[])

---

start

```
public abstract int start()
```

Starts the node detection and event generation processes for the `NMS` object.

Returns:
`0` on success

---

stop

```
public abstract int stop()
```

Stops the node detection and event generation processes for the `NMS` object.

Returns:
`0` on success

---

startMGClient

```
public abstract int startMGClient(short mode,
                                  java.lang.String server,
                                  int port,
                                  boolean useSSL,
                                  java.lang.String userName,
                                  java.lang.String password,
                                  boolean ignoreLocalPackets)
```

Starts the Meshdynamics Management Gateway client for remote management.

The Meshdynamics Management Gateway client connects to a Meshdynamics Management Gateway server using the HTTP protocol.

Parameters:

`mode` - the client mode, can be one of `MG_CLIENT_MODE_FORWARDER` or `MG_CLIENT_MODE_REMOTE_MANAGER`
`server` - the IP address or host name of the Meshdynamics Management Gateway server
`port` - the port on which the Meshdynamics Management Gateway server listens
`useSSL` - set to `true` if a SSL connection is to be used
`userName` - the account user-name at the Meshdynamics Management Gateway server
`password` - the account password
`ignoreLocalPackets` - local incoming packets will be ignored in `MG_CLIENT_MODE_REMOTE_MANAGER` mode

Returns:

`0` on success

---

stopMGClient

```
public abstract int stopMGClient()
```

Stops the Meshdynamics Management Gateway client for remote management.

Returns:

`0` on success

---

openNetwork

```
public abstract NMS.Network openNetwork(java.lang.String networkName,
                                        java.lang.String networkKey,
                                        int networkType)
```

Opens the specified mesh network.

Parameters:

`networkName` - the mesh network identifier
`networkKey` - the mesh network key
`networkType` - the network type (`NMS.NETWORK_TYPE_REGULAR` or `NMS.NETWORK_TYPE_FIPS_140_2`). For `NMS.NETWORK_TYPE_FIPS_140_2` the `networkKey` specifies a 128-bit hexstring.

Returns:

reference to the `Network` object or `null` on failure

---

closeNetwork

```
public abstract int closeNetwork(NMS.Network network)
```

Closes the specified network.

Parameters:

`network` - the mesh network to be closed

Returns:

`0` on success

---

getOpenNetworks

```
public abstract java.util.Enumeration<NMS.Network> getOpenNetworks()
```

Returns an `Enumeration` of all open `Network` objects.

Returns:
`Enumeration` of all open `Network` objects.

---

getNetworkByName

```
public abstract NMS.Network getNetworkByName(java.lang.String networkName)
```

Returns a reference to a `Network` object with the specified identifier.

Parameters:
`networkName` - the mesh network identifier
Returns:
reference to the `Network` object or `null` on failure

---

stdOutPrintln

```
public abstract void stdOutPrintln(java.lang.String str)
```

Prints the specified string to the standard output stream.

Parameters:
`str` - the string to be printed

---

stdErrPrintln

```
public abstract void stdErrPrintln(java.lang.String str)
```

Prints the specified string to the error output stream.

Parameters:
`str` - the string to be printed

---

unInitialize

```
protected abstract void unInitialize()
```

Un-initializes the `NMS` instance.

All Classes

NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfiguration
NMS.Hashtable
NMS.InterfaceConfiguration
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSecurity
NMS.WPAPersonalSecurity

Constant Field Values

Contents

- com.meshdynamics.*

com.meshdynamics.*

com.meshdynamics.api.NMS		
public static final short	CIPHER_CCMP	1
public static final short	CIPHER_TKIP	2
public static final short	COUNTRY_CODE_CUSTOM	1
public static final short	COUNTRY_CODE_DEFAULT	0
public static final short	EFFISTREAM_MATCH_ETH_DST	2
public static final short	EFFISTREAM_MATCH_ETH_SRC	3
public static final short	EFFISTREAM_MATCH_ETH_TYPE	1
public static final short	EFFISTREAM_MATCH_IGNORE	0
public static final short	EFFISTREAM_MATCH_IP_DIFFSRV	5
public static final short	EFFISTREAM_MATCH_IP_DST	7
public static final short	EFFISTREAM_MATCH_IP_PROTO	8
public static final short	EFFISTREAM_MATCH_IP_SRC	6
public static final short	EFFISTREAM_MATCH_IP_TOS	4
public static final short	EFFISTREAM_MATCH_RTP_LENGTH	17
public static final short	EFFISTREAM_MATCH_RTP_PAYLOAD	16
public static final short	EFFISTREAM_MATCH_RTP_VERSION	15
public static final short	EFFISTREAM_MATCH_TCP_DST_PORT	13
public static final short	EFFISTREAM_MATCH_TCP_LENGTH	14
public static final short	EFFISTREAM_MATCH_TCP_SRC_PORT	12
public static final short	EFFISTREAM_MATCH_UDP_DST_PORT	10
public static final short	EFFISTREAM_MATCH_UDP_LENGTH	11
public static final short	EFFISTREAM_MATCH_UDP_SRC_PORT	9
public static final int	EVENT_NETWORK_CLOSE	5
public static final int	EVENT_NODE_DEAD	3
public static final int	EVENT_NODE_HEARTBEAT	1
public static final int	EVENT_NODE_HEARTBEAT_MISS	2
public static final int	EVENT_NODE_SCAN	4
public static final short	MG_CLIENT_MODE_FORWARDER	1
public static final short	MG_CLIENT_MODE_REMOTE_MANAGER	2
public static final short	NETWORK_TYPE_FIPS_140_2	2
public static final short	NETWORK_TYPE_REGULAR	1
public static final short	OPTION_ADHOC	2
public static final short	OPTION_ADHOC_DHCP	16
public static final short	OPTION_ADHOC_INFRA_BEGIN	8

public static final short	OPTION_ADHOC_SECTORED	64
public static final short	OPTION_FORCED_ROOT	4
public static final short	OPTION_IGMP	1
public static final short	OPTION_LOCATION	32
public static final short	OPTION_SIP	128
public static final short	PERFORMANCE_PROTOCOL_TCP	1
public static final short	PERFORMANCE_PROTOCOL_UDP	2
public static final short	PERFORMANCE_TYPE_DUAL_INDIVIDUAL	2
public static final short	PERFORMANCE_TYPE_DUAL_SIMULTANEOUS	3
public static final short	PERFORMANCE_TYPE_SINGLE	1
public static final short	PHY_SUB_TYPE_802_11_A	1
public static final short	PHY_SUB_TYPE_802_11_B	2
public static final short	PHY_SUB_TYPE_802_11_BG	4
public static final short	PHY_SUB_TYPE_802_11_G	3
public static final short	PHY_SUB_TYPE_802_11_PSF	7
public static final short	PHY_SUB_TYPE_802_11_PSH	6
public static final short	PHY_SUB_TYPE_802_11_PSQ	5
public static final short	PHY_SUB_TYPE_IGNORE	0
public static final short	PHY_TYPE_802_11	1
public static final short	PHY_TYPE_ETHERNET	0
public static final short	REG_DOMAIN_CODE_CUSTOM	3
public static final short	REG_DOMAIN_CODE_ETSI	2
public static final short	REG_DOMAIN_CODE_FCC	1
public static final short	REG_DOMAIN_CODE_NONE	0
public static final short	SECURITY_TYPE_NONE	0
public static final short	SECURITY_TYPE_WEP_104	2
public static final short	SECURITY_TYPE_WEP_40	1
public static final short	SECURITY_TYPE_WPA_ENTERPRISE	4
public static final short	SECURITY_TYPE_WPA_PERSONAL	3
public static final short	SECURITY_TYPE_WPA2_ENTERPRISE	6
public static final short	SECURITY_TYPE_WPA2_PERSONAL	5
public static final short	USAGE_TYPE_DOWNLINK	1
public static final short	USAGE_TYPE_SCANNER	2
public static final short	USAGE_TYPE_UPLINK	0

com.meshdynamics.api.NMS.ACLEntry		
public static final short	INVALID_VLAN	-1

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS NEXT CLASS FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Class NMS.ACLConfiguration

```
java.lang.Object
  └ com.meshdynamics.api.NMS.ACLConfiguration
```

Enclosing class:
NMS

```
public static class NMS.ACLConfiguration
extends java.lang.Object
```

Defines the Access Control List configuration for a node.

Field Summary

NMS.ObjectArray	**entries** The array of NMS.ACLEntry objects.
short	**whiteList** Defines whether the ACL configuration entries specify a 'white-list'.

Constructor Summary

NMS.ACLConfiguration() Default constructor, initializes the object with an empty entries array and sets whiteList to 0.
NMS.ACLConfiguration(java.lang.String objectNotation) Constructs the ACLConfiguration from a object notation string.

Method Summary

void	**addEntry**(NMS.ACLEntry entry) Adds the entry into the entries array.
java.lang.String	**toObjectNotation**() Returns a string containing the object notation representation of the ACLConfiguration object.
java.lang.String	**toString**()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

whiteList

```
public short whiteList
```

Defines whether the ACL configuration entries specify a 'white-list'.

If non-zero, the entries are used as a white-list i.e clients that are not in the list shall be rejected.

---

entries

```
public NMS.ObjectArray entries
```

The array of `NMS.ACLEntry` objects.

Constructor Detail

NMS.ACLConfiguration

```
public NMS.ACLConfiguration()
```

Default constructor, initializes the object with an empty entries array and sets `whiteList` to `0`.

NMS.ACLConfiguration

```
public NMS.ACLConfiguration(java.lang.String objectNotation)
```

Constructs the `ACLConfiguration` from a object notation string.

Parameters:

`objectNotation` -

Method Detail

toString

```
public java.lang.String toString()
```

Overrides:

`toString` in class `java.lang.Object`

---

toObjectNotation

```
public java.lang.String toObjectNotation()
```

Returns a string containing the object notation representation of the ACLConfiguration object.

Returns:
the object notation string

---

addEntry

```
public void addEntry(NMS.ACLEntry entry)
```

Adds the entry into the entries array.

Parameters:
`entry` - the entry to be added

Package **Class** **Tree** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES**
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS NEXT CLASS FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Class NMS.ACLEntry

```
java.lang.Object
  └ com.meshdynamics.api.NMS.ACLEntry
```

Enclosing class:
NMS

```
public static class NMS.ACLEntry
extends java.lang.Object
```

Defines an Access Control List entry.

Field Summary

short	**block** Set to non-zero to block the device.
short	**dot11eCategory** The IEEE 802.11e access category for the device.
short	**dot11eEnabled** Set to non-zero of `dot11eCategory` is valid.
static short	**INVALID_VLAN** Constant specifying the default VLAN.
java.lang.String	**macAddress** The MAC-address of the device.
short	**vlanTag** The IEEE 802.1q VLAN tag to be used when the device associates.

Constructor Summary

NMS.ACLEntry() Default constructor.

Method Summary

java.lang.String	**toObjectNotation**() Returns a string containing the object notation representation of the `ACLEntry` object.
java.lang.String	**toString**()

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

macAddress

```
public java.lang.String macAddress
```

The MAC-address of the device.

---

vlanTag

```
public short vlanTag
```

The IEEE 802.1q VLAN tag to be used when the device associates.

Setting this value to `ACLEntry.INVALID_VLAN` will put the device on the default VLAN.

---

dot11eEnabled

```
public short dot11eEnabled
```

Set to non-zero of `dot11eCategory` is valid.

---

dot11eCategory

```
public short dot11eCategory
```

The IEEE 802.11e access category for the device.

NOTE: This field is ignored if `dot11eEnabled` is `0`.

---

block

```
public short block
```

Set to non-zero to block the device.

INVALID_VLAN

```
public static final short INVALID_VLAN
```

Constant specifying the default VLAN.

See Also:
Constant Field Values

Constructor Detail

NMS.ACLEntry

`public NMS.ACLEntry()`

Default constructor.

Method Detail

toString

`public java.lang.String toString()`

Overrides:
`toString` in class `java.lang.Object`

---

toObjectNotation

`public java.lang.String toObjectNotation()`

Returns a string containing the object notation representation of the `ACLEntry` object.

Returns:
the object notation string

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES**
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Interface NMS.ConnectedDevice

Enclosing class:
NMS

```
public static interface NMS.ConnectedDevice
```

Defines the properties of all devices connected to a `NMS.Node`

See Also:
`NMS.Node.getConnectedDevices()`

Method Summary

`java.lang.String`	**`getMacAddress`**`()` Returns the MAC address of the device formatted as a string.
`int`	**`getRxSignal`**`()` Returns the RSSI of the packets from the device to the node.
`int`	**`getTxBitRate`**`()` Returns the transmit rate of packets from the node to the device.

Method Detail

getMacAddress

```
java.lang.String getMacAddress()
```

Returns the MAC address of the device formatted as a string.

Returns:
MAC address

getRxSignal

```
int getRxSignal()
```

Returns the RSSI of the packets from the device to the node.

Returns:
signal RSSI

getTxBitRate

`int` **`getTxBitRate`**`()`

Returns the transmit rate of packets from the node to the device.

Returns:

transmit rate

Package **Class** **Tree** **Deprecated** **Index** **Help**

PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfiguratic
NMS.Hashtable
NMS.InterfaceConfigurati
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSecu
NMS.WPAPersonalSecur

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES**
SUMMARY: NESTED | FIELD | CONSTR | METHOD
DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Class NMS.EffistreamRule

```
java.lang.Object
  └ com.meshdynamics.api.NMS.EffistreamRule
```

Enclosing class:
NMS

```
public static class NMS.EffistreamRule
extends java.lang.Object
```

Defines a Effistream QoS rule.

Field Summary

short	**actionBitRate** Specifies that the transmit rate. This field is only valid for leaf-level rules.
short	**actionDot11eCategory** Specifies that the IEEE 802.11e category.
short	**actionDropPacket** Specifies that the packets will be dropped.
short	**actionNoAck** When non-zero specifies that the packets will be sent without acknowledgement.
short	**actionQueuedRetry** Specifies that the transmit rate.
NMS.EffistreamRule	**firstChild** Reference to the next child rule object.
java.lang.String	**matchCriteria** Specifies the match criteria for the rule.
short	**matchId** Specifies the match identifier for the rule.
NMS.EffistreamRule	**nextSibling** Reference to the next sibling rule object.
NMS.EffistreamRule	**parent** Reference to the parent rule object.

Constructor Summary

NMS.EffistreamRule() Default constructor typically used to create the 'ROOT' object for the rules.
NMS.EffistreamRule(short matchId, java.lang.String matchCriteria) Use this constructor to create a rule without specifying child rules.

NMS.EffistreamRule(short matchId, java.lang.String matchCriteria, NMS.EffistreamRule child) Use this constructor to create a rule directly specifying the first child.
NMS.EffistreamRule(short matchId, java.lang.String matchCriteria, short actionNoAck, short actionDropPacket, short actionDot11eCategory, short actionBitRate, short actionQueuedRetry) Use this constructor to create a leaf-level rule object.

Method Summary

void	**addChild**(NMS.EffistreamRule child) Adds a child rule to the rule object.
static NMS.EffistreamRule	**fromXmlSpec**(java.lang.String xmlSpec) Returns a EffistreamRule object hierarchy based on a XML based input.
java.lang.String	**toString**()
java.lang.String	**toXmlSpec**() Converts a EffistreamRule object hierarchy to a XML based string.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

matchId

public short **matchId**

Specifies the match identifier for the rule.

This can be one of NMS.EFFISTREAM_MATCH_ETH_DST,NMS.EFFISTREAM_MATCH_ETH_SRC, NMS.EFFISTREAM_MATCH_ETH_TYPE,NMS.EFFISTREAM_MATCH_IGNORE, NMS.EFFISTREAM_MATCH_IP_DIFFSRV,NMS.EFFISTREAM_MATCH_IP_DST, NMS.EFFISTREAM_MATCH_IP_PROTO,NMS.EFFISTREAM_MATCH_IP_SRC, NMS.EFFISTREAM_MATCH_IP_TOS,NMS.EFFISTREAM_MATCH_RTP_LENGTH, NMS.EFFISTREAM_MATCH_RTP_VERSION,NMS.EFFISTREAM_MATCH_TCP_DST_PORT, NMS.EFFISTREAM_MATCH_TCP_LENGTH,NMS.EFFISTREAM_MATCH_TCP_SRC_PORT, NMS.EFFISTREAM_MATCH_UDP_DST_PORT,NMS.EFFISTREAM_MATCH_UDP_LENGTH, NMS.EFFISTREAM_MATCH_UDP_SRC_PORT.

---

matchCriteria

public java.lang.String **matchCriteria**

Specifies the match criteria for the rule.

Depending on the value of matchId this field contains either a MAC address, an IP address, a 32-bit integer or a range of 32-bit integers all formatted as a string.

For more information on the format refer to the match identifiers :

NMS.EFFISTREAM_MATCH_ETH_DST,NMS.EFFISTREAM_MATCH_ETH_SRC,
NMS.EFFISTREAM_MATCH_ETH_TYPE,NMS.EFFISTREAM_MATCH_IGNORE,
NMS.EFFISTREAM_MATCH_IP_DIFFSRV,NMS.EFFISTREAM_MATCH_IP_DST,
NMS.EFFISTREAM_MATCH_IP_PROTO,NMS.EFFISTREAM_MATCH_IP_SRC,
NMS.EFFISTREAM_MATCH_IP_TOS,NMS.EFFISTREAM_MATCH_RTP_LENGTH,
NMS.EFFISTREAM_MATCH_RTP_VERSION,NMS.EFFISTREAM_MATCH_TCP_DST_PORT,
NMS.EFFISTREAM_MATCH_TCP_LENGTH,NMS.EFFISTREAM_MATCH_TCP_SRC_PORT,
NMS.EFFISTREAM_MATCH_UDP_DST_PORT,NMS.EFFISTREAM_MATCH_UDP_LENGTH,
NMS.EFFISTREAM_MATCH_UDP_SRC_PORT

---

actionNoAck

```
public short actionNoAck
```

When non-zero specifies that the packets will be sent without acknowledgement.

This field is only valid for leaf-level rules.

---

actionDropPacket

```
public short actionDropPacket
```

Specifies that the packets will be dropped.

This field is only valid for leaf-level rules.

---

actionDot11eCategory

```
public short actionDot11eCategory
```

Specifies that the IEEE 802.11e category.

This field is only valid for leaf-level rules.

---

actionBitRate

```
public short actionBitRate
```

Specifies that the transmit rate.
This field is only valid for leaf-level rules.

---

actionQueuedRetry

```
public short actionQueuedRetry
```

Specifies that the transmit rate.

This field is only valid for leaf-level rules.

---

parent

```
public NMS.EffistreamRule parent
```

Reference to the parent rule object.

---

nextSibling

`public` [NMS.EffistreamRule](#) **`nextSibling`**

Reference to the next sibling rule object.

---

firstChild

`public` [NMS.EffistreamRule](#) **`firstChild`**

Reference to the next child rule object.

When `null`, the rule is a leaf-level rule.

Constructor Detail

NMS.EffistreamRule

`public` **`NMS.EffistreamRule`**`()`

Default constructor typically used to create the 'ROOT' object for the rules.

NMS.EffistreamRule

```
public NMS.EffistreamRule(short matchId,
                          java.lang.String matchCriteria)
```

Use this constructor to create a rule without specifying child rules.

Parameters:

`matchId` - the match identifier for the rule see [matchId](#)
`matchCriteria` - the criteria for a match see [matchCriteria](#)

NMS.EffistreamRule

```
public NMS.EffistreamRule(short matchId,
                          java.lang.String matchCriteria,
                          NMS.EffistreamRule child)
```

Use this constructor to create a rule directly specifying the first child.

e.g. `rule = new EffistreamRule(NMS.EFFISTREAM_MATCH_ETH_TYPE,"2048",new EffistreamRule(NMS.EFFISTREAM_MATCH_IP_SRC,"192.168.45.6",0,0,3,36,0)))`

Parameters:

`matchId` - the match identifier for the rule see [matchId](#)
`matchCriteria` - the criteria for a match see [matchCriteria](#)
`child` - the first child rule [firstChild](#)

NMS.EffistreamRule

```
public NMS.EffistreamRule(short matchId,
```

```
                        java.lang.String matchCriteria,
                        short actionNoAck,
                        short actionDropPacket,
                        short actionDot11eCategory,
                        short actionBitRate,
                        short actionQueuedRetry)
```

Use this constructor to create a leaf-level rule object.

Parameters:

`matchId` - the match identifier for the rule see `matchId`
`matchCriteria` - the criteria for a match see `matchCriteria`
`actionNoAck` - see `actionNoAck`
`actionDropPacket` - see `actionDropPacket`
`actionDot11eCategory` - see `actionDot11eCategory`
`actionBitRate` - see `actionBitRate`
`actionQueuedRetry` - see `actionQueuedRetry`

Method Detail

addChild

```
public void addChild(NMS.EffistreamRule child)
```

Adds a child rule to the rule object.

The child rule is added to the tail of the siblings list

Parameters:

`child` - the child rule to add

---

toXmlSpec

```
public java.lang.String toXmlSpec()
```

Converts a EffistreamRule object hierarchy to a XML based string.

Returns:

xml based effistream rule hierarchy

---

fromXmlSpec

```
public static NMS.EffistreamRule fromXmlSpec(java.lang.String xmlSpec)
```

Returns a EffistreamRule object hierarchy based on a XML based input.

Parameters:

`xmlSpec` - the XML input string

Returns:

a EffistreamRule object hierarchy

Throws:

`java.lang.Exception`

---

toString

```
public java.lang.String toString()
```

Overrides:

`toString` in class `java.lang.Object`

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfiguration
NMS.Hashtable
NMS.InterfaceConfiguratio
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSecur
NMS.WPAPersonalSecurit

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS NEXT CLASS
FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD
DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Class NMS.GeneralConfiguration

```
java.lang.Object
  └ com.meshdynamics.api.NMS.GeneralConfiguration
```

Enclosing class:
NMS

```
public static class NMS.GeneralConfiguration
extends java.lang.Object
```

Defines all Node level fields used by a `NMS.Node`.

See Also:
`NMS.Node.getGeneralConfiguration()`,
`NMS.Node.setGeneralConfiguration(com.meshdynamics.api.NMS.GeneralConfiguration)`

Field Summary

int	**countryCode** The operating country code for the node.
short	**dfsRequired** Specifies whether Dynamics Frequency Selection and RADAR detection is required for the `regulatoryDomain`.
short	**dynamicChannelAllocation** The node level flag determining whether downlink interfaces shall use the Dynamic Channel Allocation scheme.
java.lang.String	**gatewayIpAddress** The ip-address of the default gateway in dotted decimal form.
java.lang.String	**gpsLatitude** Latitude coordinate of the node in decimal format.
java.lang.String	**gpsLongitude** Longitude coordinate of the node in decimal format.
short	**heartbeatInterval** The heartbeat interval for the node.
java.lang.String	**hostName** The network host-name for the node.
java.lang.String	**ipAddress** The ip-address for the node in dotted decimal form.
short	**mobilityMode** The node's mobilty mode.
java.lang.String	**model** The model identifier for the node.
java.lang.String	**nodeDescription** User-defined description for the node
java.lang.String	**nodeName** User-defined name of the node
short	

	options The combination of run-time options enabled on the node.
java.lang.String	**preferredParent** The MAC address of the preferred parent's downlink radio.
int	**regulatoryDomain** The operating regulatory domain for the node.
java.lang.String	**subnetMask** The subnet-mask for the node in dotted decimal form.

Constructor Summary

NMS.GeneralConfiguration()

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

nodeName

public java.lang.String **nodeName**

User-defined name of the node

---

nodeDescription

public java.lang.String **nodeDescription**

User-defined description for the node

---

model

public java.lang.String **model**

The model identifier for the node.

NOTE: This field is read-only and will be ignored in calls to NMS.Node.setGeneralConfiguration(com.meshdynamics.api.NMS.GeneralConfiguration).

---

gpsLatitude

public java.lang.String **gpsLatitude**

Latitude coordinate of the node in decimal format.

Coordinates South of the equator are represented by a negative number

---

gpsLongitude

`public java.lang.String` **`gpsLongitude`**

Longitude coordinate of the node in decimal format.

Coordinates West of the meridian are represented by a negative number

---

hostName

`public java.lang.String` **`hostName`**

The network host-name for the node.

---

ipAddress

`public java.lang.String` **`ipAddress`**

The ip-address for the node in dotted decimal form.

---

subnetMask

`public java.lang.String` **`subnetMask`**

The subnet-mask for the node in dotted decimal form.

---

gatewayIpAddress

`public java.lang.String` **`gatewayIpAddress`**

The ip-address of the default gateway in dotted decimal form.

---

preferredParent

`public java.lang.String` **`preferredParent`**

The MAC address of the preferred parent's downlink radio.

---

heartbeatInterval

`public short` **`heartbeatInterval`**

The heartbeat interval for the node.

---

mobilityMode

`public short` **`mobilityMode`**

The node's mobilty mode.

A non-zero value indicates that the node is configured for mobility.

---

options

`public short` **`options`**

The combination of run-time options enabled on the node.

See Also:

NMS.OPTION_ADHOC, NMS.OPTION_ADHOC_DHCP, NMS.OPTION_ADHOC_INFRA_BEGIN, NMS.OPTION_ADHOC_SECTORED, NMS.OPTION_FORCED_ROOT, NMS.OPTION_IGMP, NMS.OPTION_LOCATION, NMS.OPTION_SIP

---

dynamicChannelAllocation

```
public short dynamicChannelAllocation
```

The node level flag determining whether downlink interfaces shall use the Dynamic Channel Allocation scheme.

A value of 0 will turn off the dynamic channel allocation scheme even if it is turned on for individual downlink interfaces.

---

countryCode

```
public int countryCode
```

The operating country code for the node.

A value of 0 indicates the default country code.

---

regulatoryDomain

```
public int regulatoryDomain
```

The operating regulatory domain for the node.

See Also:

NMS.REG_DOMAIN_CODE_NONE, NMS.REG_DOMAIN_CODE_CUSTOM, NMS.REG_DOMAIN_CODE_FCC, NMS.REG_DOMAIN_CODE_ETSI

---

dfsRequired

```
public short dfsRequired
```

Specifies whether Dynamics Frequency Selection and RADAR detection is required for the `regulatoryDomain`.

Constructor Detail

NMS.GeneralConfiguration

```
public NMS.GeneralConfiguration()
```

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES**
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Class NMS.Hashtable

```
java.lang.Object
  └ com.meshdynamics.api.NMS.Hashtable
```

Enclosing class:
NMS

```
public static class NMS.Hashtable
extends java.lang.Object
```

The Hashtable class provides an implementation of a Hashtable of generic 'Object' keys and generic 'Object' values.

Constructor Summary

`NMS.Hashtable``()` Default constructor.

Method Summary

`void`	**`clear`**`()` Clears the hashtable.
`java.lang.Object`	**`get`**`(java.lang.Object key)` Retrieves the value for the specified key.
`java.util.Enumeration<java.lang.Object>`	**`keys`**`()` Returns an `Enumeration` of all the keys in the hashtable.
`void`	**`put`**`(java.lang.Object key, java.lang.Object value)` Inserts the specified value for the specified key into the hashtable.
`void`	**`remove`**`(java.lang.Object key)` Removes the specified key from the hashtable.

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

NMS.Hashtable

```
public NMS.Hashtable()
```

Default constructor.

Method Detail

get

```
public java.lang.Object get(java.lang.Object key)
```

Retrieves the value for the specified key.

Parameters:
`key` - the key for which the value is to be retrieved
Returns:
the value

---

put

```
public void put(java.lang.Object key,
                java.lang.Object value)
```

Inserts the specified value for the specified key into the hashtable.

Parameters:
`key` - the key for which the value is to be inserted
`value` - the value to be inserted

---

remove

```
public void remove(java.lang.Object key)
```

Removes the specified key from the hashtable.

---

clear

```
public void clear()
```

Clears the hashtable.

---

keys

```
public java.util.Enumeration<java.lang.Object> keys()
```

Returns an `Enumeration` of all the keys in the hashtable.

Returns:

`Enumeration` object for the keys

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfiguration
NMS.Hashtable
NMS.InterfaceConfiguration
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSecurit
NMS.WPAPersonalSecurity

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS NEXT CLASS FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Class NMS.InterfaceConfiguration

```
java.lang.Object
  └ com.meshdynamics.api.NMS.InterfaceConfiguration
```

Enclosing class:
NMS

```
public static class NMS.InterfaceConfiguration
extends java.lang.Object
```

Defines the interface level settings for a `NMS.Node`.

See Also:
`NMS.Node.getInterfaces()`, `NMS.Node.getInterfaceConfigurationByName(java.lang.String)`

Field Summary

int	**ackTimeout** The timeout in multiples of `10` micro-seconds for the 802.11 ACK frame.
short	**allowClientConnection** When set to `0`, does not allow standard 802.11 clients to connect to the interface, and only allows child mesh nodes to connect to the interface.
NMS.ShortArray	**dcaList** When `dynamicChannelAllocation` is non-zero, downlink interfaces choose the best channel from the integers specified in this array.
short	**dynamicChannelAllocation** When set to `0`, disables dynamic channel allocation and forces the interface to use the channel specified by `manualChannel`.
java.lang.String	**essid** The ESSID used in 802.11 beacons and 802.11 probe-response packets transmitted by the downlink interface.
int	**fragThreshold** The 802.11 fragmentation threshold for the interface.
short	**hideEssid** When non-zero causes the ESSID field of 802.11 beacons and broadcast probe-responses to contain an empty string.
short	**identifier** The identifier for the interface.
java.lang.String	**macAddress** The MAC address of the interface.
short	**manualChannel** The channel to be used when `dynamicChannelAllocation` is set to 0.
int	**maxTransmitRate** The maximum transmit rate for the interface.
java.lang.String	**name** The name of the interface.
short	**operatingChannel** The current operating channel for the interface.

short	**phySubType** Defines the physical layer sub-type used by the interface.
short	**phyType** Defines the Physical layer used by the interface.
int	**rtsThreshold** The 802.11 RTS threshold for the interface.
java.lang.Object	**securityInfo** Opaque object containing the security settings for the interface.
short	**securityType** The encryption/authentication scheme used to secure connections on the interface.
int	**transmitPower** The transmit power for the interface.
short	**usageType** Defines the role in which the interface is used during the node's operation.

Constructor Summary

NMS.InterfaceConfiguration() Default constructor.
NMS.InterfaceConfiguration(java.lang.String objectNotation) Initializes the configuration from the object notation string.

Method Summary

java.lang.String	**toObjectNotation**() Returns a string containing the object notation representation for the interface.
java.lang.String	**toString**()

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

name

public java.lang.String **name**

The name of the interface.

---

macAddress

public java.lang.String **macAddress**

The MAC address of the interface.

NOTE: The value of this field will be ignored in calls to NMS.Node.setInterfaceConfiguration(com.meshdynamics.api.NMS.InterfaceConfiguration).

---

identifier

public short **identifier**

The identifier for the interface.

The interfaces of a node are identified according to the `usageType` and `phySubType` fields.

e.g. For a node with two 802.11a downlinks and a 802.11g downlink, the first downlink shall have an identifier of 0, while the 2nd will have 1.

The 802.11g downlink will have an identifier of 0.

NOTE: The value of this field will be ignored in calls to `NMS.Node.setInterfaceConfiguration(com.meshdynamics.api.NMS.InterfaceConfiguration)`.

---

usageType

`public short` **`usageType`**

Defines the role in which the interface is used during the node's operation.

NOTE: The value of this field will be ignored in calls to `NMS.Node.setInterfaceConfiguration(com.meshdynamics.api.NMS.InterfaceConfiguration)`.

See Also:

`NMS.USAGE_TYPE_DOWNLINK`, `NMS.USAGE_TYPE_UPLINK`, `NMS.USAGE_TYPE_SCANNER`

---

phyType

`public short` **`phyType`**

Defines the Physical layer used by the interface.

NOTE: The value of this field will be ignored in calls to `NMS.Node.setInterfaceConfiguration(com.meshdynamics.api.NMS.InterfaceConfiguration)`.

See Also:

`NMS.PHY_TYPE_ETHERNET`, `NMS.PHY_TYPE_802_11`

---

phySubType

`public short` **`phySubType`**

Defines the physical layer sub-type used by the interface.

See Also:

`NMS.PHY_SUB_TYPE_IGNORE`, `NMS.PHY_SUB_TYPE_802_11_A`, `NMS.PHY_SUB_TYPE_802_11_B`, `NMS.PHY_SUB_TYPE_802_11_G`, `NMS.PHY_SUB_TYPE_802_11_BG`, `NMS.PHY_SUB_TYPE_802_11_PSQ`, `NMS.PHY_SUB_TYPE_802_11_PSH`, `NMS.PHY_SUB_TYPE_802_11_PSF`

---

essid

`public java.lang.String` **`essid`**

The ESSID used in 802.11 beacons and 802.11 probe-response packets transmitted by the downlink interface.

This field is ignored for 802.11 uplink, scanner interfaces.

For ETHERNET downlinks, this field specifies the VLAN configuration for the ethernet port :

- ESSID of a VLAN - only allows the specified VLAN
- MD-PRIV-SSID-NO-VLAN - No VLANs allowed.
- Other - All VLANs allowed

---

hideEssid

`public short` **`hideEssid`**

When non-zero causes the ESSID field of 802.11 beacons and broadcast probe-responses to contain an empty string.

This field is ignored for 802.11 uplink, scanner interfaces and by all ethernet interfaces.

---

maxTransmitRate

`public int` **`maxTransmitRate`**

The maximum transmit rate for the interface.

When set to `0`, the interface uses all the transmit rates defined by the physical layer sub-type.

This field is ignored for ethernet interfaces.

---

transmitPower

`public int` **`transmitPower`**

The transmit power for the interface.

This field is ignored for ethernet interfaces.

---

ackTimeout

`public int` **`ackTimeout`**

The timeout in multiples of `10` micro-seconds for the 802.11 ACK frame.

Transmissions with the ACK frame not arriving within the ackTimeout value are considered erroneous and are retried.

This field is ignored for ethernet interfaces.

---

allowClientConnection

`public short` **`allowClientConnection`**

When set to `0`, does not allow standard 802.11 clients to connect to the interface, and only allows child mesh nodes to connect to the interface.

This field is ignored for ethernet interfaces.

---

fragThreshold

`public int` **`fragThreshold`**

The 802.11 fragmentation threshold for the interface.

All packets larger than the fragThreshold shall be fragmented.

This field is ignored for ethernet interfaces.

---

rtsThreshold

`public int` **`rtsThreshold`**

The 802.11 RTS threshold for the interface.

All packets larger than the rtsThreshold shall be preceeded by the standard 802.11 RTS/CTS mechanism to ensure error free reception.

This field is ignored for ethernet interfaces.

---

dynamicChannelAllocation

`public short` **`dynamicChannelAllocation`**

When set to `0`, disables dynamic channel allocation and forces the interface to use the channel specified by `manualChannel`.

When set to a non-zero value, the interface chooses the best channel from the `dcaList` for operation.

This field is ignored for 802.11 uplink scanner and ethernet interfaces.

See Also:
`dcaList`, `manualChannel`

---

manualChannel

`public short` **`manualChannel`**

The channel to be used when `dynamicChannelAllocation` is set to 0.

This field is ignored for 802.11 uplink scanner and ethernet interfaces.

See Also:
`dynamicChannelAllocation`

---

dcaList

`public NMS.ShortArray` **`dcaList`**

When `dynamicChannelAllocation` is non-zero, downlink interfaces choose the best channel from the integers specified in this array.

For uplink interfaces, if the list is empty, all channels shall be scanned. If the list is non-empty only the channels specified in the list will be scanned for parent selection.

NOTE: The list must not be empty for uplink interfaces if the node is in disjoint-adhoc mode.

For scanner interfaces, the list determines the channels that will be scanned for detecting prospective parent nodes.

This field is ignored for ethernet interfaces.

See Also:
`dynamicChannelAllocation`

---

securityType

`public short` **`securityType`**

The encryption/authentication scheme used to secure connections on the interface.

This field is ignored for 802.11 uplink scanner and ethernet interfaces.

See Also:
NMS.SECURITY_TYPE_NONE, NMS.SECURITY_TYPE_WEP_104, NMS.SECURITY_TYPE_WEP_40, NMS.SECURITY_TYPE_WPA2_ENTERPRISE, NMS.SECURITY_TYPE_WPA2_PERSONAL, NMS.SECURITY_TYPE_WPA_ENTERPRISE, NMS.SECURITY_TYPE_WPA_PERSONAL

---

securityInfo

```
public java.lang.Object securityInfo
```

Opaque object containing the security settings for the interface.

The field represents a NMS.WEPSecurity object if securityType is NMS.SECURITY_TYPE_WEP_104 or NMS.SECURITY_TYPE_WEP_40.

The field represents a NMS.WPAPersonalSecurity object if securityType is NMS.SECURITY_TYPE_WPA2_PERSONAL or NMS.SECURITY_TYPE_WPA_PERSONAL.

The field represents a NMS.WPAEnterpriseSecurity object if securityType is NMS.SECURITY_TYPE_WPA2_ENTERPRISE or NMS.SECURITY_TYPE_WPA_ENTERPRISE.

This field is ignored for 802.11 uplink scanner and ethernet interfaces.

See Also:
securityType, NMS.WEPSecurity, NMS.WPAPersonalSecurity, NMS.WPAEnterpriseSecurity

---

operatingChannel

```
public short operatingChannel
```

The current operating channel for the interface.

Constructor Detail

NMS.InterfaceConfiguration

```
public NMS.InterfaceConfiguration()
```

Default constructor.

NMS.InterfaceConfiguration

```
public NMS.InterfaceConfiguration(java.lang.String objectNotation)
```

Initializes the configuration from the object notation string.

Parameters:
objectNotation - the object notation string

Method Detail

toString

```
public java.lang.String toString()
```

Overrides:
toString in class java.lang.Object

---

toObjectNotation

```
public java.lang.String toObjectNotation()
```

Returns a string containing the object notation representation for the interface.

Returns:

string containing object notation representation of the interface

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS NEXT CLASS FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Interface NMS.NeighborNode

Enclosing class:
NMS

`public static interface` **`NMS.NeighborNode`**

Defines the properties of all neighbor nodes detected by a `NMS.Node`

See Also:
`NMS.Node.getNeighborNodes()`

Method Summary

`int`	**`getDownlinkCount`**`()` Returns the number of downlink radios seen by the node.
`NMS.Node`	**`getNode`**`()` Returns a reference to the `NMS.Node` object representing the neighbor.
`int`	**`getUplinkSignal`**`()` Returns the RSSI of the neighbor's first downlink signal as seen by the uplink.
`int`	**`getUplinkSignal`**`(int downlinkIndex)` Returns the RSSI as seen by the uplink from the specific downlink of the neighbor.
`int`	**`getUplinkTxBitRate`**`()` Returns the transmit rate from the uplink to the neighbor's first downlink.
`int`	**`getUplinkTxBitRate`**`(int downlinkIndex)` Returns the transmit rate from the uplink to the specfic downlink of the neighbor.

Method Detail

getNode

`NMS.Node` **`getNode`**`()`

Returns a reference to the `NMS.Node` object representing the neighbor.

Returns:
a reference to the `NMS.Node` object representing the neighbor

---

getUplinkSignal

`int` **`getUplinkSignal`**`()`

Returns the RSSI of the neighbor's first downlink signal as seen by the uplink.

Returns:
signal RSSI
See Also:
`getUplinkSignal(int)`

---

getUplinkTxBitRate

`int` **`getUplinkTxBitRate`**`()`

Returns the transmit rate from the uplink to the neighbor's first downlink.

Returns:
transmit rate
See Also:
`getUplinkTxBitRate(int)`

---

getUplinkSignal

`int` **`getUplinkSignal`**`(int downlinkIndex)`

Returns the RSSI as seen by the uplink from the specific downlink of the neighbor.

Parameters:
`downlinkIndex` - the index of the neighbor's downlink
Returns:
signal RSSI

---

getUplinkTxBitRate

`int` **`getUplinkTxBitRate`**`(int downlinkIndex)`

Returns the transmit rate from the uplink to the specfic downlink of the neighbor.

Parameters:
`downlinkIndex` - the index of the neighbor's downlink
Returns:
transmit rate

---

getDownlinkCount

`int` **`getDownlinkCount`**`()`

Returns the number of downlink radios seen by the node.

Returns:

downlink count

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfiguration
NMS.Hashtable
NMS.InterfaceConfiguratio
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSecur
NMS.WPAPersonalSecurit

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS NEXT CLASS FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Interface NMS.Network

Enclosing class:
NMS

```
public static interface NMS.Network
```

The `Network` interface defines all properties and actions associated with a mesh network.

A mesh network is a community of mesh nodes that can :

- Communicate with each other using a common security parameters.
- Be managed as a single entity.

Method Summary

int	**addListener**(NMS.NetworkListener networklistener) Adds the specified NetworkListener callback hook to the mesh network.
int	**deleteNode**(NMS.Node node) Deletes the specified node from the mesh network.
java.lang.String	**getName**() Returns the name of the mesh network.
NMS.Node	**getNodeByMacAddress**(java.lang.String macAddress) Returns the Node object representing the specified MAC-address.
java.util.Enumeration<NMS.Node>	**getNodes**() Returns an Enumeration of all mesh nodes in the network.
int	**removeListener**(NMS.NetworkListener networklistener) Removes the specified NetworkListener callback hook from the mesh network.
int	**waitForNodeDetect**(java.lang.String macAddresses, long timeout) Blocks the calling thread until all the nodes specified in macAddresses parameter are fully detected and configurable.

Method Detail

getName

```
java.lang.String getName()
```

Returns the name of the mesh network.

Returns:
`String` object containing the name of the mesh network

---

getNodes

`java.util.Enumeration<NMS.Node> getNodes()`

Returns an `Enumeration` of all mesh nodes in the network.

Returns:
`Enumeration` of all mesh nodes in the network.
See Also:
`NMS.Node`

---

deleteNode

`int deleteNode(NMS.Node node)`

Deletes the specified node from the mesh network.

Parameters:
`node` - the node to be deleted
Returns:
`0` if successful

---

addListener

`int addListener(NMS.NetworkListener networklistener)`

Adds the specified `NetworkListener` callback hook to the mesh network.

The `NetworkListener` callback hook enables the caller to receive information on the events that occur in the mesh network.

Parameters:
`networklistener` - the `NetworkListener` callback hook to be added
Returns:
`0` if successful
See Also:
`NMS.NetworkListener`

---

removeListener

`int removeListener(NMS.NetworkListener networklistener)`

Removes the specified `NetworkListener` callback hook from the mesh network.

If successful, the caller will no longer be able to receive information on the events that occur in the mesh network.

Parameters:
`networklistener` - the `NetworkListener` callback hook to be removed
Returns:
`0` if successful
See Also:
`NMS.NetworkListener`

---

getNodeByMacAddress

`NMS.Node getNodeByMacAddress(java.lang.String macAddress)`

Returns the `Node` object representing the specified MAC-address.

Parameters:

`macAddress` - the mesh node's unit MAC-address to be searched

Returns:

`Node` object representing the specified MAC-address.

See Also:

`NMS.Node`

---

waitForNodeDetect

```
int waitForNodeDetect(java.lang.String macAddresses,
                      long timeout)
```

Blocks the calling thread until all the nodes specified in `macAddresses` parameter are fully detected and configurable.

Parameters:

`macAddresses` - A string containing comma-seperated list of MAC-addresses to detect

`timeout` - the number of milli-seconds to block until nodes get detected

Returns:

`0` if successful or negative integer if a timeout occurs.

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES**
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Interface NMS.NetworkListener

Enclosing class:
NMS

`public static interface` **`NMS.NetworkListener`**

The `NetworkListener` interface is used to receive events on a mesh network.

See Also:
`NMS.Network.addListener(com.meshdynamics.api.NMS.NetworkListener)`

Method Summary

`int`	**`onEvent`**`(int event, NMS.Network network, NMS.Node node)` This method is called when an event occurs on the network.

Method Detail

onEvent

```
int onEvent(int event,
            NMS.Network network,
            NMS.Node node)
```

This method is called when an event occurs on the network.

Parameters:

`event` - the code specifying the event that occurred. It can be one of the following:

`NMS.EVENT_NODE_DEAD`,`NMS.EVENT_NODE_HEARTBEAT`,
`NMS.EVENT_NODE_HEARTBEAT_MISS`,`NMS.EVENT_NODE_SCAN`

`network` - the network on which the event occurred
`node` - the node for which the event occurred

Returns:

Currently the return value is ignored and must be set to 0

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES**
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfiguration
NMS.Hashtable
NMS.InterfaceConfiguration
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSecurity
NMS.WPAPersonalSecurity

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS NEXT CLASS
FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD
DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Interface NMS.Node

Enclosing class:
NMS

```
public static interface NMS.Node
```

The Node interface defines all the properties and actions that can be carried out on a mesh node.

Method Summary

int	**addVlan**(NMS.VlanConfiguration configuration) Adds the specified VLAN to the Node.
int	**beginConfigurationUpdate**() Starts a configuration transaction bracket.
int	**cancelConfigurationUpdate**() Closes the current configuration transaction bracket without sending the configuration update.
int	**commitConfigurationUpdate**() Closes the current configuration transaction bracket and sends the updated configuration to the Node.
java.lang.String	**executeCommand**(java.lang.String command) Executes a Meshdynamics MeshCommand™ on the Node.
java.lang.String	**generateConfigMacro**(java.lang.String scriptLanguage) Generates a configuration macro script for the Node.
NMS.ACLConfiguration	**getACLConfiguration**() Returns the Access Control List configuration for the Node.
java.util.Enumeration<NMS.ConnectedDevice>	**getConnectedDevices**() Returns an Enumeration of devices that are connected to this Node.
short	**getCpuUsage**() Returns the current average CPU usage for the node.
NMS.EffistreamRule	**getEffistreamRules**() Returns the Effistream™ rule hierarchy for the Node.
short	**getFirmwareVersionMajor**() Returns the major firmware version for the Node.
short	**getFirmwareVersionMinor**() Returns the minor firmware version for the Node.
short	**getFirmwareVersionVariant**() Returns the firmware version variant for the Node.
short	**getFreeRAM**() Returns the amount of free RAM in Mega-bytes.
NMS.GeneralConfiguration	**getGeneralConfiguration**() Returns the node level configuration of the Node.
short	**getGpsAltitude**() Returns the current operational altitude in meters.
java.lang.String	**getGpsCurrentLatitude**() Returns the current operational latitude coordinate in decimal format.
java.lang.String	**getGpsCurrentLongitude**() Returns the current operational longitude coordinate in decimal format.
short	**getGpsSpeed**() Returns the current operational speed in Km/Hr.
long	**getHeartbeatSqnr**() Returns the sequence number of the last heartbeat received from the node.
short	**getHopCount**() Returns the current hop level for the node.
short	**getInputVoltage**() Returns the current input voltage to the node.
NMS.InterfaceConfiguration	**getInterfaceConfigurationByName**(java.lang.String name)

	Returns the configuration of the specified interface.
java.util.Enumeration<NMS.InterfaceConfiguration>	**getInterfaces**() Returns an Enumeration of all interfaces in the Node.
java.util.Enumeration<NMS.NeighborNode>	**getNeighborNodes**() Returns an Enumeration of nodes that this Node sees as neighbors.
java.lang.String	**getParentBssid**() Returns the MAC-address of the parent's downlink on which this Node is connected.
int	**getParentDownlinkSignal**() Returns the signal RSSI in packets received by the parent's downlink interface from this node's uplink.
int	**getParentDownlinkTxBitRate**() Returns the transmit rate used by the parent for packet's transmitted to this Node.
short	**getTemperature**() Returns the current node enclosure temperature.
short	**getTreeLinkRate**() Returns the 'Tree Link Rate' for the node.
java.lang.String	**getUnitMacAddress**() Returns the MAC address of the node formatted as a string.
NMS.VlanConfiguration	**getVlanConfigurationByTag**(short tag) Returns the configuration of the specified VLAN.
java.util.Enumeration<NMS.VlanConfiguration>	**getVlans**() Returns an Enumeration of all VLANS in the Node.
short	**isIpReachable**() Returns non-zero if this Node can be communicated with using IP.
boolean	**isMobile**() Returns whether the node is mobile or stationary.
boolean	**isRemote**() Returns whether the remote or local.
void	**reboot**() REBOOT's the Node.
short	**rebootRequired**() Returns non-zero if a 'REBOOT' is required for the Node.
int	**removeVlan**(short tag) Removes the specified VLAN from the Node.
int	**restoreDefaults**() Restore's the Node to factory configuration.
java.lang.String	**runPerformanceTest**(int recordCount, short type, short protocol, int udpBandWidth) Provides network performance information to the Node.
int	**setACLConfiguration**(NMS.ACLConfiguration configuration) Sets the Node's Access Control List configuration.
int	**setEffistreamRules**(NMS.EffistreamRule rules) Updates the Effistream™ rule hierarchy for the Node.
int	**setGeneralConfiguration**(NMS.GeneralConfiguration configuration) Updates the node level configuration for the Node.
int	**setInterfaceConfiguration**(NMS.InterfaceConfiguration configuration) Updates the interface configuration for the Node.
int	**setVlanConfiguration**(NMS.VlanConfiguration configuration) Sets the configuration of an existing VLAN in the Node.
int	**setVlans**(NMS.ObjectArray vlans) Sets the Node's VLAN list from a ObjectArray.
int	**upgradeFirmware**(java.lang.String firmwareFilePath) Upgrades the firmware of the Node.

Method Detail

getUnitMacAddress

java.lang.String **getUnitMacAddress**()

Returns the MAC address of the node formatted as a string.

Returns:
MAC address

---

getHeartbeatSqnr

```
long getHeartbeatSqnr()
```

Returns the sequence number of the last heartbeat received from the node.

Returns:
heartbeat sequence number

---

isMobile

```
boolean isMobile()
```

Returns whether the node is mobile or stationary.

Returns:
`true` if the node is mobile, `false` otherwise

---

isRemote

```
boolean isRemote()
```

Returns whether the remote or local.

Returns:
`true` if node is remote, `false` otherwise

---

getFreeRAM

```
short getFreeRAM()
```

Returns the amount of free RAM in Mega-bytes.

Returns:
free RAM in Mega-bytes

---

getInputVoltage

```
short getInputVoltage()
```

Returns the current input voltage to the node.

Returns:
node input voltage

---

getTreeLinkRate

```
short getTreeLinkRate()
```

Returns the 'Tree Link Rate' for the node.

The 'Tree Link Rate' is the lowest rate in the path from the node to the ROOT.

Returns:
the 'Tree Link Rate'

---

getHopCount

```
short getHopCount()
```

Returns the current hop level for the node.

Returns:
the number of hops away from the ROOT.

---

getCpuUsage

```
short getCpuUsage()
```

Returns the current average CPU usage for the node.

Returns:
the average cpu usage as a percentage

---

getTemperature

```
short getTemperature()
```

Returns the current node enclosure temperature.

Returns:
the current temperature inside the node enclosure in Celcius.

---

getParentDownlinkSignal

```
int getParentDownlinkSignal()
```

Returns the signal RSSI in packets received by the parent's downlink interface from this node's uplink.

Returns:
the signal RSSI received by the parent's downlink interface.

---

getParentDownlinkTxBitRate

```
int getParentDownlinkTxBitRate()
```

Returns the transmit rate used by the parent for packet's transmitted to this `Node`.

Returns:
the transmit rate for packets transmitted by parent's downlink.

---

getParentBssid

```
java.lang.String getParentBssid()
```

Returns the MAC-address of the parent's downlink on which this `Node` is connected.

Returns:
MAC-address of parent's downlink interface

---

getGpsCurrentLatitude

```
java.lang.String getGpsCurrentLatitude()
```

Returns the current operational latitude coordinate in decimal format.

Coordinates South of the equator are represented by a negative number.

Returns:
the current operational latitude coordinate

---

getGpsCurrentLongitude

```
java.lang.String getGpsCurrentLongitude()
```

Returns the current operational longitude coordinate in decimal format.

Coordinates West of the meridian are represented by a negative number.

Returns:
the current operational longitude coordinate

---

getGpsSpeed

```
short getGpsSpeed()
```

Returns the current operational speed in Km/Hr.

Returns:
the current operational speed

---

getGpsAltitude

`short` **`getGpsAltitude`**`()`

Returns the current operational altitude in meters.

Returns:
the the current operational altitude in meters

---

getFirmwareVersionMajor

`short` **`getFirmwareVersionMajor`**`()`

Returns the major firmware version for the `Node`.

Returns:
the major firmware version.

---

getFirmwareVersionMinor

`short` **`getFirmwareVersionMinor`**`()`

Returns the minor firmware version for the `Node`.

Returns:
the minor firmware version.

---

getFirmwareVersionVariant

`short` **`getFirmwareVersionVariant`**`()`

Returns the firmware version variant for the `Node`.

Returns:
the firmware version variant.

---

isIpReachable

`short` **`isIpReachable`**`()`

Returns non-zero if this `Node` can be communicated with using IP.

Returns:
`0` if node is not IP-reachable.
See Also:
`NMS.GeneralConfiguration.ipAddress`

---

rebootRequired

`short` **`rebootRequired`**`()`

Returns non-zero if a 'REBOOT' is required for the `Node`.

Returns:
`0` if the changes to the `Node`'s configuration dot not require a reboot. `non-zero` if a reboot is required.

---

getNeighborNodes

`java.util.Enumeration<NMS.NeighborNode>` **`getNeighborNodes`**`()`

Returns an `Enumeration` of nodes that this `Node` sees as neighbors.

Neighbor nodes are pottential parent nodes, and are connected to, in the event of a link failure.

Returns:
`Enumeration` of `NeighborNode` objects

---

getConnectedDevices

`java.util.Enumeration<NMS.ConnectedDevice>` **`getConnectedDevices`**`()`

Returns an `Enumeration` of devices that are connected to this `Node`.

This method returns standard client devices and child mesh nodes.

Returns:
`Enumeration` of `ConnectedDevice` objects

---

getGeneralConfiguration

`NMS.GeneralConfiguration` **`getGeneralConfiguration`**`()`

Returns the node level configuration of the `Node`.

Returns:
the node level configuration of the `Node`

---

getInterfaces

`java.util.Enumeration<NMS.InterfaceConfiguration>` **`getInterfaces`**`()`

Returns an `Enumeration` of all interfaces in the `Node`.

Returns:
`Enumeration` of `InterfaceConfiguration` objects

---

getVlans

`java.util.Enumeration<NMS.VlanConfiguration>` **`getVlans`**`()`

Returns an `Enumeration` of all VLANS in the `Node`.

Returns:
`Enumeration` of `VlanConfiguration` objects

---

getInterfaceConfigurationByName

`NMS.InterfaceConfiguration` **`getInterfaceConfigurationByName`**`(java.lang.String name)`

Returns the configuration of the specified interface.

Parameters:
`name` - the name of the interface
Returns:
`InterfaceConfiguration` object for the interface

---

getVlanConfigurationByTag

`NMS.VlanConfiguration` **`getVlanConfigurationByTag`**`(short tag)`

Returns the configuration of the specified VLAN.

Parameters:
`tag` - the VLAN identifier
Returns:
`VlanConfiguration` object for the VLAN

---

getEffistreamRules

`NMS.EffistreamRule` **`getEffistreamRules`**`()`

Returns the Effistream™ rule hierarchy for the `Node`.

Returns:
`EffistreamRule` object hierachy

---

getACLConfiguration

`NMS.ACLConfiguration` **`getACLConfiguration`**`()`

Returns the Access Control List configuration for the `Node`.

Returns:
`ACLConfiguration` object

---

reboot

```
void reboot()
```

REBOOT's the `Node`.

---

restoreDefaults

```
int restoreDefaults()
```

Restore's the `Node` to factory configuration.

Returns:
`0` on success

---

executeCommand

```
java.lang.String executeCommand(java.lang.String command)
```

Executes a Meshdynamics MeshCommand™ on the `Node`.

Parameters:
`command` - the Meshdynamics MeshCommand™ to execute
Returns:
the result of the command

---

upgradeFirmware

```
int upgradeFirmware(java.lang.String firmwareFilePath)
```

Upgrades the firmware of the `Node`.

The firmware file must be one that is created specifically for the MAC address of the `Node`.

Parameters:
`firmwareFilePath` - the path to the firmware upgrade file.
Returns:
`0` on success

---

runPerformanceTest

```
java.lang.String runPerformanceTest(int recordCount,
                                    short type,
                                    short protocol,
                                    int udpBandWidth)
```

Provides network performance information to the `Node`.

The performance test is run from the host to the `Node` and hence will reflect the network performance of all links along the path.

Parameters:
`recordCount` - the number of performanc records to be run
`type` - the type of the performance run, can be one of `NMS.PERFORMANCE_TYPE_SINGLE`, `NMS.PERFORMANCE_TYPE_DUAL_INDIVIDUAL`, `NMS.PERFORMANCE_TYPE_DUAL_SIMULTANEOUS`
`protocol` - the protocol to be used, can be one of `NMS.PERFORMANCE_PROTOCOL_TCP`, `NMS.PERFORMANCE_PROTOCOL_UDP`.
`udpBandWidth` - when using `PERFORMANCE_PROTOCOL_UDP`, the bandwidth in Kbps.
Returns:
the result of the performance test

---

setGeneralConfiguration

```
int setGeneralConfiguration(NMS.GeneralConfiguration configuration)
```

Updates the node level configuration for the `Node`.

If `beginConfigurationUpdate` has been called prior to this method, the updated configuration will be sent upon a call to the method `commitConfigurationUpdate`.

If `beginConfigurationUpdate` has not been called prior to this method, the configuration is sent immedietly.

Parameters:
`configuration` - the node level configuration
Returns:
`0` upon success

---

setInterfaceConfiguration

`int` **`setInterfaceConfiguration`**`(`[`NMS.InterfaceConfiguration`]` configuration)`

Updates the interface configuration for the `Node`.

The interface is specified by the `name` field of the `InterfaceConfiguration` object.

If `beginConfigurationUpdate` has been called prior to this method, the updated configuration will be sent upon a call to the method `commitConfigurationUpdate`.

If `beginConfigurationUpdate` has not been called prior to this method, the configuration is sent immediatly.

Parameters:
`configuration` - the configuration for the interface
Returns:
`0` upon success

---

setEffistreamRules

`int` **`setEffistreamRules`**`(`[`NMS.EffistreamRule`]` rules)`

Updates the Effistream™ rule hierarchy for the `Node`.

If `beginConfigurationUpdate` has been called prior to this method, the updated configuration will be sent upon a call to the method `commitConfigurationUpdate`.

If `beginConfigurationUpdate` has not been called prior to this method, the configuration is sent immediatly.

Parameters:
`rules` - the Effistream™ rule hierarchy
Returns:
`0` upon success

---

addVlan

`int` **`addVlan`**`(`[`NMS.VlanConfiguration`]` configuration)`

Adds the specified VLAN to the `Node`.

If `beginConfigurationUpdate` has been called prior to this method, the updated configuration will be sent upon a call to the method `commitConfigurationUpdate`.

If `beginConfigurationUpdate` has not been called prior to this method, the configuration is sent immediatly.

Parameters:
`configuration` - the `VlanConfiguration` object
Returns:
`0` upon success

---

setVlanConfiguration

`int` **`setVlanConfiguration`**`(`[`NMS.VlanConfiguration`]` configuration)`

Sets the configuration of an existing VLAN in the `Node`.

The `essid` and `tag` fields of the `VlanConfiguration` object are used to identify the existing VLAN.

If no existing VLAN exisits, the method returns an error.

If `beginConfigurationUpdate` has been called prior to this method, the updated configuration will be sent upon a call to the method `commitConfigurationUpdate`.

If `beginConfigurationUpdate` has not been called prior to this method, the configuration is sent immediatly.

Parameters:
`configuration` - the `VlanConfiguration` object
Returns:
`0` upon success

---

removeVlan

`int` **`removeVlan`**`(short tag)`

Removes the specified VLAN from the `Node`.

The tag field is used to identify the VLAN.

If no existing VLAN exisits, the method returns an error.

If `beginConfigurationUpdate` has been called prior to this method, the updated configuration will be sent upon a call to the method `commitConfigurationUpdate`.

If `beginConfigurationUpdate` has not been called prior to this method, the configuration is sent immedietly.

Parameters:
`tag` - the tag to identify the existing VLAN
Returns:
`0` upon success

---

setVlans

`int` **`setVlans`**`(`NMS.ObjectArray `vlans)`

Sets the `Node`'s VLAN list from a `ObjectArray`.

This method delete's all existing VLANs and adds all VLANs in the `ObjectArray`.

If `beginConfigurationUpdate` has been called prior to this method, the updated configuration will be sent upon a call to the method `commitConfigurationUpdate`.

If `beginConfigurationUpdate` has not been called prior to this method, the configuration is sent immedietly.

Parameters:
`vlans` - `ObjectArray` containing `VlanConfiguration` objects
Returns:
`0` upon success

---

setACLConfiguration

`int` **`setACLConfiguration`**`(`NMS.ACLConfiguration `configuration)`

Sets the `Node`'s Access Control List configuration.

This method delete's all existing entries from the ACL configuration and sets the `Node`'s Access Control List configuration as specified by the `ACLConfiguration` object.

If `beginConfigurationUpdate` has been called prior to this method, the updated configuration will be sent upon a call to the method `commitConfigurationUpdate`.

If `beginConfigurationUpdate` has not been called prior to this method, the configuration is sent immedietly.

Parameters:
`configuration` - the `ACLConfiguration` object
Returns:
`0` upon success

---

generateConfigMacro

`java.lang.String` **`generateConfigMacro`**`(java.lang.String scriptLanguage)`

Generates a configuration macro script for the `Node`.

Parameters:
`scriptLanguage` - the scripting lanugage to use
Returns:
string containing the configuration macro script

---

beginConfigurationUpdate

`int` **`beginConfigurationUpdate`**`()`

Starts a configuration transaction bracket.

After a call to this method, calls that update the `Node`'s configuration are be not be sent immedietly, but are deferred until a call to `commitConfigurationUpdate`.

The configuration transaction bracket can be closed by a call to `commitConfigurationUpdate` or to `cancelConfigurationUpdate`.

Returns:
`0` upon success

cancelConfigurationUpdate

`int` **`cancelConfigurationUpdate`**`()`

Closes the current configuration transaction bracket without sending the configuration update.

Returns:
`0` upon success

commitConfigurationUpdate

`int` **`commitConfigurationUpdate`**`()`

Closes the current configuration transaction bracket and sends the updated configuration to the `Node`.

Returns:
`0` upon success

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES**
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Class NMS.ObjectArray

```
java.lang.Object
  └ com.meshdynamics.api.NMS.ObjectArray
```

Enclosing class:
NMS

```
public static class NMS.ObjectArray
extends java.lang.Object
```

The ObjectArray class provides an interface to a growable array that stores object references.

Constructor Summary

NMS.ObjectArray()	Default constructor to create the array with 0 elements.
NMS.ObjectArray(int length)	Constructor to create the array with specified number of elements initialized to null.

Method Summary

void	**add**(java.lang.Object value) Add a object reference to the end of the array and increase the length by 1.
void	**clear**() Removes all elements in the array and sets the number of elements to 0.
java.lang.Object	**get**(int index) Retrieves the object reference at the specified index.
int	**length**() Retrieve the number of elements in the ObjectArray.
void	**removeAt**(int index) Removes the element at the specified index.
void	**set**(int index, java.lang.Object value) Set the object reference at the specified index.
java.lang.String	**toObjectNotation**() Returns a string containing the object notation representation for the ObjectArray.
java.lang.String	**toString**()

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

NMS.ObjectArray

```
public NMS.ObjectArray()
```

Default constructor to create the array with 0 elements.

NMS.ObjectArray

```
public NMS.ObjectArray(int length)
```

Constructor to create the array with specified number of elements initialized to null.

Method Detail

set

```
public void set(int index,
                java.lang.Object value)
```

Set the object reference at the specified index.

Parameters:

`index` - the index
`value` - the object reference

---

get

```
public java.lang.Object get(int index)
```

Retrieves the object reference at the specified index.

Parameters:

`index` - the index

Returns:

the object reference

---

length

```
public int length()
```

Retrieve the number of elements in the `ObjectArray`.

Returns:
the number of elements

---

removeAt

```
public void removeAt(int index)
```

Removes the element at the specified index.

Parameters:
`index` - the index of the element to be removed.

---

add

```
public void add(java.lang.Object value)
```

Add a object reference to the end of the array and increase the length by 1.

Parameters:
`value` - the object reference to be added

---

clear

```
public void clear()
```

Removes all elements in the array and sets the number of elements to 0.

---

toString

```
public java.lang.String toString()
```

Overrides:
`toString` in class `java.lang.Object`

---

toObjectNotation

```
public java.lang.String toObjectNotation()
```

Returns a string containing the object notation representation for the `ObjectArray`.

Returns:
string containing object notation

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS NEXT CLASS FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Class NMS.ShortArray

```
java.lang.Object
  └ com.meshdynamics.api.NMS.ShortArray
```

Enclosing class:
NMS

```
public static class NMS.ShortArray
extends java.lang.Object
```

Defines an array of short integers.

Constructor Summary

NMS.ShortArray(int length) Constructs ShortArray object with specified number of elements.
NMS.ShortArray(short... numbers) Constructs ShortArray object with the specified elements.
NMS.ShortArray(java.lang.String values) Constructs ShortArray object from a comma seperated list of numbers.

Method Summary

short	**get**(int index) Retrieve the value at the specified index.
int	**length**() Retrieve the number of elements in the ShortArray.
void	**set**(int index, short value) Set the value at specified index.
void	**set**(short... numbers) Set the elements of the ShortArray to the specified variable argument list of numbers.
void	**set**(java.lang.String values) Set the elements of the ShortArray from a comma seperated list of numbers.
java.lang.String	**toObjectNotation**() Returns a string containing the object notation representation for the ShortArray.
java.lang.String	**toString**()

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

NMS.ShortArray

```
public NMS.ShortArray(int length)
```

Constructs `ShortArray` object with specified number of elements.

Parameters:

`length` - the number of elements

NMS.ShortArray

```
public NMS.ShortArray(short... numbers)
```

Constructs `ShortArray` object with the specified elements.

Parameters:

`numbers` - variable argument list of short inetegers

NMS.ShortArray

```
public NMS.ShortArray(java.lang.String values)
```

Constructs `ShortArray` object from a comma seperated list of numbers.

Parameters:

`values` - string containing comma seperated list of numbers

Method Detail

set

```
public void set(short... numbers)
```

Set the elements of the `ShortArray` to the specified variable argument list of numbers.

Parameters:

`numbers` - variable argument list of short inetegers

---

set

```
public void set(java.lang.String values)
```

Set the elements of the `ShortArray` from a comma seperated list of numbers.

Parameters:
`values` - string specifying comma seperated list of values

---

set

```
public void set(int index,
                short value)
```

Set the value at specified index.

Parameters:
`index` - the index
`value` - the value

---

get

```
public short get(int index)
```

Retrieve the value at the specified index.

Parameters:
`index` - the index
Returns:
the value at the specified index

---

length

```
public int length()
```

Retrieve the number of elements in the `ShortArray`.

Returns:
the number of elements

---

toString

```
public java.lang.String toString()
```

Overrides:
`toString` in class `java.lang.Object`

---

toObjectNotation

```
public java.lang.String toObjectNotation()
```

Returns a string containing the object notation representation for the `ShortArray`.

Returns:

string containing object notation representation

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS NEXT CLASS FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Class NMS.Thread

```
java.lang.Object
  └java.lang.Thread
      └com.meshdynamics.api.NMS.Thread
```

All Implemented Interfaces:
java.lang.Runnable

Enclosing class:
NMS

```
public static class NMS.Thread
extends java.lang.Thread
```

The Thread class provides multi-threading functionality to scripting platforms.

Nested Class Summary

static interface	**NMS.Thread.Runnable** The Runnable interface is implemented by any class whose instances are executed by a thread.

Nested classes/interfaces inherited from class java.lang.Thread
java.lang.Thread.State, java.lang.Thread.UncaughtExceptionHandler

Field Summary

Fields inherited from class java.lang.Thread
MAX_PRIORITY, MIN_PRIORITY, NORM_PRIORITY

Constructor Summary

NMS.Thread(NMS.Thread.Runnable runnable) Default constructor

Method Summary

void	**run**()
static void	**sleep**(long milliSeconds)

	The `sleep` method blocks the calling thread for the specified number of milli-seconds. Since it is a static method, the calling thread does not have to be an instance of the `NMS.Thread` class.
`void`	**`start`**`()` Starts the thread.

Methods inherited from class java.lang.Thread
`activeCount, checkAccess, countStackFrames, currentThread, destroy, dumpStack, enumerate, getAllStackTraces, getContextClassLoader, getDefaultUncaughtExceptionHandler, getId, getName, getPriority, getStackTrace, getState, getThreadGroup, getUncaughtExceptionHandler, holdsLock, interrupt, interrupted, isAlive, isDaemon, isInterrupted, join, join, join, resume, setContextClassLoader, setDaemon, setDefaultUncaughtExceptionHandler, setName, setPriority, setUncaughtExceptionHandler, sleep, stop, stop, suspend, toString, yield`

Methods inherited from class java.lang.Object
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructor Detail

NMS.Thread

`public` **`NMS.Thread`**`(`NMS.Thread.Runnable` runnable)`

Default constructor

Parameters:

`runnable` - the reference to an object implementing the `Runnable` interface

Method Detail

sleep

`public static void` **`sleep`**`(long milliSeconds)`

The `sleep` method blocks the calling thread for the specified number of milli-seconds.

Since it is a static method, the calling thread does not have to be an instance of the `NMS.Thread` class.

Parameters:

`milliSeconds` - the number of milli-seconds to block

---

start

`public void` **`start`**`()`

Starts the thread.

Overrides:

`start` in class `java.lang.Thread`

---

run

```
public void run()
```

Specified by:

`run` in interface `java.lang.Runnable`

Overrides:

`run` in class `java.lang.Thread`

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES**
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Interface NMS.Thread.Runnable

Enclosing class:
NMS.Thread

`public static interface` **`NMS.Thread.Runnable`**

The `Runnable` interface is implemented by any class whose instances are executed by a thread.

The interface defines a single method `run` that represents the running thread.

See Also:
`NMS.Thread`

Method Summary

`void`	**`run`**`()` The `run` method implements the logic for the thread.

Method Detail

run

`void` **`run`**`()`

The `run` method implements the logic for the thread.

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES**
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS NEXT CLASS FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Class NMS.VlanConfiguration

```
java.lang.Object
  └ com.meshdynamics.api.NMS.VlanConfiguration
```

Enclosing class:
NMS

```
public static class NMS.VlanConfiguration
extends java.lang.Object
```

Defines the settings for a Virtual-LAN in a `NMS.Node`.

Field Summary

short	**dot11eCategory** The IEEE 802.11e access category to be used for packets for the VLAN.
short	**dot11eEnabled** Non-zero if IEEE 802.11e based QOS is enabled for the VLAN.
short	**dot1pPriority** The IEEE 802.1p bridge priority for the VLAN.
java.lang.String	**essid** The ESSID used in 802.11 probe-response packets.
java.lang.String	**name** The friendly name for the VLAN.
java.lang.Object	**securityInfo** Opaque object containing the security settings for the VLAN.
short	**securityType** The encryption/authentication scheme used to secure connections on the VLAN.
short	**tag** The IEEE 802.1q tag for the VLAN.

Constructor Summary

NMS.VlanConfiguration() Default constructor.
NMS.VlanConfiguration(java.lang.String objectNotation) Creates a VlanConfiguration object from a object notation string.

Method Summary

java.lang.String	**toObjectNotation**() Returns a string containing the object notation representation of the VlanConfiguration object.
java.lang.String	**toString**()

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

name

```
public java.lang.String name
```

The friendly name for the VLAN.

---

essid

```
public java.lang.String essid
```

The ESSID used in 802.11 probe-response packets.

---

tag

```
public short tag
```

The IEEE 802.1q tag for the VLAN.

---

dot11eEnabled

```
public short dot11eEnabled
```

Non-zero if IEEE 802.11e based QOS is enabled for the VLAN.

---

dot11eCategory

```
public short dot11eCategory
```

The IEEE 802.11e access category to be used for packets for the VLAN.

Ignored if `dot11eEnabled` is `0`.

---

dot1pPriority

`public short` **`dot1pPriority`**

The IEEE 802.1p bridge priority for the VLAN.

---

securityType

`public short` **`securityType`**

The encryption/authentication scheme used to secure connections on the VLAN.

See Also:

`NMS.SECURITY_TYPE_NONE`, `NMS.SECURITY_TYPE_WEP_104`, `NMS.SECURITY_TYPE_WEP_40`, `NMS.SECURITY_TYPE_WPA2_ENTERPRISE`, `NMS.SECURITY_TYPE_WPA2_PERSONAL`, `NMS.SECURITY_TYPE_WPA_ENTERPRISE`, `NMS.SECURITY_TYPE_WPA_PERSONAL`

---

securityInfo

`public java.lang.Object` **`securityInfo`**

Opaque object containing the security settings for the VLAN.

The field represents a `NMS.WEPSecurity` object if `securityType` is `NMS.SECURITY_TYPE_WEP_104` or `NMS.SECURITY_TYPE_WEP_40`.

The field represents a `NMS.WPAPersonalSecurity` object if `securityType` is `NMS.SECURITY_TYPE_WPA2_PERSONAL` or `NMS.SECURITY_TYPE_WPA_PERSONAL`.

The field represents a `NMS.WPAEnterpriseSecurity` object if `securityType` is `NMS.SECURITY_TYPE_WPA2_ENTERPRISE` or `NMS.SECURITY_TYPE_WPA_ENTERPRISE`.

This field is ignored for 802.11 uplink scanner and ethernet interfaces.

See Also:

`securityType`, `NMS.WEPSecurity`, `NMS.WPAPersonalSecurity`, `NMS.WPAEnterpriseSecurity`

Constructor Detail

NMS.VlanConfiguration

`public` **`NMS.VlanConfiguration`**`()`

Default constructor.

NMS.VlanConfiguration

`public` **`NMS.VlanConfiguration`**`(java.lang.String objectNotation)`

Creates a VlanConfiguration object from a object notation string.

Parameters:

`objectNotation` - the object notation string

Method Detail

toString

`public java.lang.String **toString**()`

Overrides:

`toString` in class `java.lang.Object`

---

toObjectNotation

`public java.lang.String **toObjectNotation**()`

Returns a string containing the object notation representation of the VlanConfiguration object.

Returns:

the object notation string

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES**
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Class NMS.WEPSecurity

```
java.lang.Object
  └ com.meshdynamics.api.NMS.WEPSecurity
```

Enclosing class:
NMS

```
public static class NMS.WEPSecurity
extends java.lang.Object
```

Defines the information used by the IEEE 802.11 **Wired Equivalent Privacy** (WEP) setting by a Node's downlink interface.

See Also:
`NMS.InterfaceConfiguration.securityType`, `NMS.InterfaceConfiguration.securityInfo`

Field Summary

short	**keyIndex** The index of the key used for transmitting packets.
NMS.ObjectArray	**wepKeys** An array of upto 4 WEP keys formatted as hexadecimal strings.

Constructor Summary

NMS.WEPSecurity() Default constructor.

Method Summary

java.lang.String	**toObjectNotation**() Returns a string containing the object notation representation of the `WEPSecurity` object
java.lang.String	**toString**()

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Field Detail

keyIndex

`public short` **`keyIndex`**

The index of the key used for transmitting packets.

For WEP-40 the valid values are 0-3.

For WEP-104 the value is ignored.

---

wepKeys

`public` `NMS.ObjectArray` **`wepKeys`**

An array of upto 4 WEP keys formatted as hexadecimal strings.

When using WEP-40 the array shall contain 4 entries of 10 hexadecimal digits.

For WEP-104 the array shall contain 1 entry of 26 hexadecimal digits

Constructor Detail

NMS.WEPSecurity

`public` **`NMS.WEPSecurity`**`()`

Default constructor.

Method Detail

toString

`public java.lang.String` **`toString`**`()`

Overrides:

`toString` in class `java.lang.Object`

---

toObjectNotation

`public java.lang.String` **`toObjectNotation`**`()`

Returns a string containing the object notation representation of the `WEPSecurity` object

Returns:

the object notation string

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS NEXT CLASS FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Class NMS.WPAEnterpriseSecurity

```
java.lang.Object
  └ com.meshdynamics.api.NMS.WPAEnterpriseSecurity
```

Enclosing class:
NMS

```
public static class NMS.WPAEnterpriseSecurity
extends java.lang.Object
```

Defines the information used for the Wifi Protected Access security setting by a Node's downlink interface in an enterprise environment.

See Also:
`NMS.InterfaceConfiguration.securityType`, `NMS.InterfaceConfiguration.securityInfo`

Field Summary

short	**cipherType** Defines the encryption mechanism to be used.
java.lang.String	**radiusServerIp** IP-address of the RADIUS server
short	**radiusServerPort** The UDP port used by the RADIUS server
java.lang.String	**radiusServerSecret** The secret key used to authenticate RADIUS packets sent by the node

Constructor Summary

NMS.WPAEnterpriseSecurity() Default constructor

Method Summary

java.lang.String	**toObjectNotation**() Returns a string containing the object notation representation of the WPAEnterpriseSecurity object.
java.lang.String	**toString**()

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

radiusServerIp

`public java.lang.String` **`radiusServerIp`**

IP-address of the RADIUS server

---

radiusServerPort

`public short` **`radiusServerPort`**

The UDP port used by the RADIUS server

---

radiusServerSecret

`public java.lang.String` **`radiusServerSecret`**

The secret key used to authenticate RADIUS packets sent by the node

---

cipherType

`public short` **`cipherType`**

Defines the encryption mechanism to be used.

See Also:
NMS.CIPHER_CCMP, NMS.CIPHER_TKIP

Constructor Detail

NMS.WPAEnterpriseSecurity

`public` **`NMS.WPAEnterpriseSecurity`**`()`

Default constructor

Method Detail

toString

`public java.lang.String` **`toString`**`()`

Overrides:

`toString` in class `java.lang.Object`

---

toObjectNotation

`public java.lang.String` **`toObjectNotation`**`()`

Returns a string containing the object notation representation of the `WPAEnterpriseSecurity` object.

Returns:

the object notation string

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package **Class** **Tree** **Deprecated** **Index** **Help**
PREV CLASS NEXT CLASS **FRAMES** **NO FRAMES**
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

com.meshdynamics.api

Class NMS.WPAPersonalSecurity

```
java.lang.Object
  └ com.meshdynamics.api.NMS.WPAPersonalSecurity
```

Enclosing class:
NMS

```
public static class NMS.WPAPersonalSecurity
extends java.lang.Object
```

Defines the information used for the Wifi Protected Access (WPA) security setting by a node's downlink interface.

See Also:
`NMS.InterfaceConfiguration.securityType`, `NMS.InterfaceConfiguration.securityInfo`

Field Summary

short	**cipherType** Defines the encryption mechanism to be used.
java.lang.String	**preSharedKey** The 256-bit pre-shared key (PSK) formatted as a hexidecimal string.

Constructor Summary

NMS.WPAPersonalSecurity() Default constructor

Method Summary

java.lang.String	**toObjectNotation**() Returns a string containing the object notation representation of the `WPAPersonalSecurity` object
java.lang.String	**toString**()

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Field Detail

preSharedKey

```
public java.lang.String preSharedKey
```

The 256-bit pre-shared key (PSK) formatted as a hexidecimal string.

The string shall consist of 64 hexadecimal digits.

---

cipherType

```
public short cipherType
```

Defines the encryption mechanism to be used.

See Also:
NMS.CIPHER_CCMP, NMS.CIPHER_TKIP

Constructor Detail

NMS.WPAPersonalSecurity

```
public NMS.WPAPersonalSecurity()
```

Default constructor

Method Detail

toString

```
public java.lang.String toString()
```

Overrides:
toString in class java.lang.Object

---

toObjectNotation

```
public java.lang.String toObjectNotation()
```

Returns a string containing the object notation representation of the WPAPersonalSecurity object

Returns:
the object notation string

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package Class **Tree** **Deprecated** **Index** **Help**
PREV NEXT **FRAMES** **NO FRAMES**

Hierarchy For Package com.meshdynamics.api

Class Hierarchy

- java.lang.Object
  - com.meshdynamics.api.**NMS**
  - com.meshdynamics.api.**NMS.ACLConfiguration**
  - com.meshdynamics.api.**NMS.ACLEntry**
  - com.meshdynamics.api.**NMS.EffistreamRule**
  - com.meshdynamics.api.**NMS.GeneralConfiguration**
  - com.meshdynamics.api.**NMS.Hashtable**
  - com.meshdynamics.api.**NMS.InterfaceConfiguration**
  - com.meshdynamics.api.**NMS.ObjectArray**
  - com.meshdynamics.api.**NMS.ShortArray**
  - com.meshdynamics.api.**NMS.VlanConfiguration**
  - com.meshdynamics.api.**NMS.WEPSecurity**
  - com.meshdynamics.api.**NMS.WPAEnterpriseSecurity**
  - com.meshdynamics.api.**NMS.WPAPersonalSecurity**
  - java.lang.Thread (implements java.lang.Runnable)
    - com.meshdynamics.api.**NMS.Thread**

Interface Hierarchy

- com.meshdynamics.api.**NMS.ConnectedDevice**
- com.meshdynamics.api.**NMS.NeighborNode**
- com.meshdynamics.api.**NMS.Network**
- com.meshdynamics.api.**NMS.NetworkListener**
- com.meshdynamics.api.**NMS.Node**
- com.meshdynamics.api.**NMS.Thread.Runnable**

Package Class **Tree** **Deprecated** **Index** **Help**
PREV NEXT **FRAMES** **NO FRAMES**

All Classes

NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Deprecated API

Contents

Package Class **Tree** **Deprecated** **Index** **Help**
PREV NEXT **FRAMES** **NO FRAMES**

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package Class **Tree** **Deprecated** **Index** **Help**
PREV NEXT FRAMES NO FRAMES
A B C D E F G H I K L M N O P R S T U V W

A

ackTimeout - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The timeout in multiples of `10` micro-seconds for the 802.11 ACK frame.
actionBitRate - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Specifies that the transmit rate.
This field is only valid for leaf-level rules.
actionDot11eCategory - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Specifies that the IEEE 802.11e category.
actionDropPacket - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Specifies that the packets will be dropped.
actionNoAck - Variable in class com.meshdynamics.api.NMS.EffistreamRule
When non-zero specifies that the packets will be sent without acknowledgement.
actionQueuedRetry - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Specifies that the transmit rate.
add(Object) - Method in class com.meshdynamics.api.NMS.ObjectArray
Add a object reference to the end of the array and increase the length by 1.
addChild(NMS.EffistreamRule) - Method in class com.meshdynamics.api.NMS.EffistreamRule
Adds a child rule to the rule object.
addEntry(NMS.ACLEntry) - Method in class com.meshdynamics.api.NMS.ACLConfiguration
Adds the entry into the entries array.
addListener(NMS.NetworkListener) - Method in interface com.meshdynamics.api.NMS.Network
Adds the specified `NetworkListener` callback hook to the mesh network.
addVlan(NMS.VlanConfiguration) - Method in interface com.meshdynamics.api.NMS.Node
Adds the specified VLAN to the `Node`.
allowClientConnection - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
When set to `0`, does not allow standard 802.11 clients to connect to the interface, and only allows child mesh nodes to connect to the interface.

B

beginConfigurationUpdate() - Method in interface com.meshdynamics.api.NMS.Node
Starts a configuration transaction bracket.
block - Variable in class com.meshdynamics.api.NMS.ACLEntry
Set to non-zero to block the device.
bytesToHexString(byte[]) - Static method in class com.meshdynamics.api.NMS
This utility method converts a byte array to a hexadecimal string.

C

cancelConfigurationUpdate() - Method in interface com.meshdynamics.api.NMS.Node
Closes the current configuration transaction bracket without sending the configuration update.
CIPHER_CCMP - Static variable in class com.meshdynamics.api.NMS
Specifies AES-CCMP encryption to be used for WPA/WPA2 Personal/Enterprise security options.
CIPHER_TKIP - Static variable in class com.meshdynamics.api.NMS
Specifies TKIP encryption to be used for WPA/WPA2 Personal/Enterprise security options.
cipherType - Variable in class com.meshdynamics.api.NMS.WPAEnterpriseSecurity
Defines the encryption mechanism to be used.
cipherType - Variable in class com.meshdynamics.api.NMS.WPAPersonalSecurity
Defines the encryption mechanism to be used.
clear() - Method in class com.meshdynamics.api.NMS.Hashtable
Clears the hashtable.
clear() - Method in class com.meshdynamics.api.NMS.ObjectArray
Removes all elements in the array and sets the number of elements to 0.
closeNetwork(NMS.Network) - Method in class com.meshdynamics.api.NMS
Closes the specified network.
com.meshdynamics.api - package com.meshdynamics.api

commitConfigurationUpdate() - Method in interface com.meshdynamics.api.NMS.Node
Closes the current configuration transaction bracket and sends the updated configuration to the `Node`.
COUNTRY_CODE_CUSTOM - Static variable in class com.meshdynamics.api.NMS
Specifies the use of custom channels.
COUNTRY_CODE_DEFAULT - Static variable in class com.meshdynamics.api.NMS
Specifies the default country code for node operation.
countryCode - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The operating country code for the node.

D

dcaList - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
When `dynamicChannelAllocation` is non-zero, downlink interfaces choose the best channel from the integers specified in this array.
deleteNode(NMS.Node) - Method in interface com.meshdynamics.api.NMS.Network
Deletes the specified node from the mesh network.
dfsRequired - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
Specifies whether Dynamics Frequency Selection and RADAR detection is required for the `regulatoryDomain`.
dot11eCategory - Variable in class com.meshdynamics.api.NMS.ACLEntry
The IEEE 802.11e access category for the device.
dot11eCategory - Variable in class com.meshdynamics.api.NMS.VlanConfiguration
The IEEE 802.11e access category to be used for packets for the VLAN.
dot11eEnabled - Variable in class com.meshdynamics.api.NMS.ACLEntry
Set to non-zero of `dot11eCategory` is valid.
dot11eEnabled - Variable in class com.meshdynamics.api.NMS.VlanConfiguration
Non-zero if IEEE 802.11e based QOS is enabled for the VLAN.
dot1pPriority - Variable in class com.meshdynamics.api.NMS.VlanConfiguration
The IEEE 802.1p bridge priority for the VLAN.

dynamicChannelAllocation - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The node level flag determining whether downlink interfaces shall use the Dynamic Channel Allocation scheme.
dynamicChannelAllocation - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
When set to `0`, disables dynamic channel allocation and forces the interface to use the channel specified by `manualChannel`.

E

EFFISTREAM_MATCH_ETH_DST - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the ETHERNET destination address field. The `matchCriteria` of the `EffistreamRule` specifies a string containing a MAC-address.
EFFISTREAM_MATCH_ETH_SRC - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the ETHERNET source address field. The `matchCriteria` of the `EffistreamRule` specifies a string containing a MAC-address.
EFFISTREAM_MATCH_ETH_TYPE - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the ETHERNET type field.
EFFISTREAM_MATCH_IGNORE - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code used at the ROOT level. The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.
EFFISTREAM_MATCH_IP_DIFFSRV - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the IP Diffrentiated services field. The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.
EFFISTREAM_MATCH_IP_DST - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the IP destination address field. The `matchCriteria` of the `EffistreamRule` specifies a string containing a IP-address.
EFFISTREAM_MATCH_IP_PROTO - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the IP protocol field. The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.
EFFISTREAM_MATCH_IP_SRC - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the IP source address field. The `matchCriteria` of the `EffistreamRule` specifies a string containing a IP-address.
EFFISTREAM_MATCH_IP_TOS - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the IP Type-of-Service field. The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.
EFFISTREAM_MATCH_RTP_LENGTH - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the RTP data length. The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).
EFFISTREAM_MATCH_RTP_PAYLOAD - Static variable in class com.meshdynamics.api.NMS
Specifies a Effistream™ match code for the RTP payload code field.

The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.

EFFISTREAM_MATCH_RTP_VERSION - Static variable in class com.meshdynamics.api.NMS

Specifies a Effistream™ match code for the RTP version field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing an integer.

EFFISTREAM_MATCH_TCP_DST_PORT - Static variable in class com.meshdynamics.api.NMS

Specifies a Effistream™ match code for the TCP destination port field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

EFFISTREAM_MATCH_TCP_LENGTH - Static variable in class com.meshdynamics.api.NMS

Specifies a Effistream™ match code for the TCP segment length.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

EFFISTREAM_MATCH_TCP_SRC_PORT - Static variable in class com.meshdynamics.api.NMS

Specifies a Effistream™ match code for the TCP source port field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

EFFISTREAM_MATCH_UDP_DST_PORT - Static variable in class com.meshdynamics.api.NMS

Specifies a Effistream™ match code for the UDP destination port field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

EFFISTREAM_MATCH_UDP_LENGTH - Static variable in class com.meshdynamics.api.NMS

Specifies a Effistream™ match code for the UDP datagram length.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

EFFISTREAM_MATCH_UDP_SRC_PORT - Static variable in class com.meshdynamics.api.NMS

Specifies a Effistream™ match code for the UDP source port field.
The `matchCriteria` of the `EffistreamRule` specifies a string containing a range (two integers seperated by a `:`).

entries - Variable in class com.meshdynamics.api.NMS.ACLConfiguration

The array of `NMS.ACLEntry` objects.

essid - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration

The ESSID used in 802.11 beacons and 802.11 probe-response packets transmitted by the downlink interface.

essid - Variable in class com.meshdynamics.api.NMS.VlanConfiguration

The ESSID used in 802.11 probe-response packets.

EVENT_NETWORK_CLOSE - Static variable in class com.meshdynamics.api.NMS

Specifies that a network was closed.

EVENT_NODE_DEAD - Static variable in class com.meshdynamics.api.NMS

Specifies that a node in unreachable in the mesh network.

EVENT_NODE_HEARTBEAT - Static variable in class com.meshdynamics.api.NMS

Specifies that a heartbeat was received from a node in the mesh network.

EVENT_NODE_HEARTBEAT_MISS - Static variable in class com.meshdynamics.api.NMS

Specifies that a node's heartbeat was missed in the mesh network.

EVENT_NODE_SCAN - Static variable in class com.meshdynamics.api.NMS

Specifies that a node is conducting dynamic channel allocation scan.

executeCommand(String) - Method in interface com.meshdynamics.api.NMS.Node
Executes a Meshdynamics MeshCommand™ on the `Node`.

F

firstChild - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Reference to the next child rule object.

fragThreshold - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The 802.11 fragmentation threshold for the interface.

fromXmlSpec(String) - Static method in class com.meshdynamics.api.NMS.EffistreamRule
Returns a EffistreamRule object hierarchy based on a XML based input.

G

gatewayIpAddress - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The ip-address of the default gateway in dotted decimal form.

generateConfigMacro(String) - Method in interface com.meshdynamics.api.NMS.Node
Generates a configuration macro script for the `Node`.

get(Object) - Method in class com.meshdynamics.api.NMS.Hashtable
Retrieves the value for the specified key.

get(int) - Method in class com.meshdynamics.api.NMS.ObjectArray
Retrieves the object reference at the specified index.

get(int) - Method in class com.meshdynamics.api.NMS.ShortArray
Retrieve the value at the specified index.

getACLConfiguration() - Method in interface com.meshdynamics.api.NMS.Node
Returns the Access Control List configuration for the `Node`.

getConnectedDevices() - Method in interface com.meshdynamics.api.NMS.Node
Returns an `Enumeration` of devices that are connected to this `Node`.

getCpuUsage() - Method in interface com.meshdynamics.api.NMS.Node
Returns the current average CPU usage for the node.

getDownlinkCount() - Method in interface com.meshdynamics.api.NMS.NeighborNode
Returns the number of downlink radios seen by the node.

getEffistreamRules() - Method in interface com.meshdynamics.api.NMS.Node
Returns the Effistream™ rule hierarchy for the `Node`.

getFirmwareVersionMajor() - Method in interface com.meshdynamics.api.NMS.Node
Returns the major firmware version for the `Node`.

getFirmwareVersionMinor() - Method in interface com.meshdynamics.api.NMS.Node
Returns the minor firmware version for the `Node`.

getFirmwareVersionVariant() - Method in interface com.meshdynamics.api.NMS.Node
Returns the firmware version variant for the `Node`.

getFreeRAM() - Method in interface com.meshdynamics.api.NMS.Node
Returns the amount of free RAM in Mega-bytes.

getGeneralConfiguration() - Method in interface com.meshdynamics.api.NMS.Node
Returns the node level configuration of the `Node`.

getGpsAltitude() - Method in interface com.meshdynamics.api.NMS.Node
Returns the current operational altitude in meters.

getGpsCurrentLatitude() - Method in interface com.meshdynamics.api.NMS.Node
Returns the current operational latitude coordinate in decimal format.

getGpsCurrentLongitude() - Method in interface com.meshdynamics.api.NMS.Node

Returns the current operational longitude coordinate in decimal format.
getGpsSpeed() - Method in interface com.meshdynamics.api.NMS.Node
Returns the current operational speed in Km/Hr.
getHeartbeatSqnr() - Method in interface com.meshdynamics.api.NMS.Node
Returns the sequence number of the last heartbeat received from the node.
getHopCount() - Method in interface com.meshdynamics.api.NMS.Node
Returns the current hop level for the node.
getInputVoltage() - Method in interface com.meshdynamics.api.NMS.Node
Returns the current input voltage to the node.
getInstance() - Static method in class com.meshdynamics.api.NMS
Returns a reference to the singleton instance of the `NMS` class.
getInterfaceConfigurationByName(String) - Method in interface com.meshdynamics.api.NMS.Node
Returns the configuration of the specified interface.
getInterfaces() - Method in interface com.meshdynamics.api.NMS.Node
Returns an `Enumeration` of all interfaces in the `Node`.
getMacAddress() - Method in interface com.meshdynamics.api.NMS.ConnectedDevice
Returns the MAC address of the device formatted as a string.
getName() - Method in interface com.meshdynamics.api.NMS.Network
Returns the name of the mesh network.
getNeighborNodes() - Method in interface com.meshdynamics.api.NMS.Node
Returns an `Enumeration` of nodes that this `Node` sees as neighbors.
getNetworkByName(String) - Method in class com.meshdynamics.api.NMS
Returns a reference to a `Network` object with the specified identifier.
getNode() - Method in interface com.meshdynamics.api.NMS.NeighborNode
Returns a reference to the `NMS.Node` object representing the neighbor.
getNodeByMacAddress(String) - Method in interface com.meshdynamics.api.NMS.Network
Returns the `Node` object representing the specified MAC-address.
getNodes() - Method in interface com.meshdynamics.api.NMS.Network
Returns an `Enumeration` of all mesh nodes in the network.
getOpenNetworks() - Method in class com.meshdynamics.api.NMS
Returns an `Enumeration` of all open `Network` objects.
getParentBssid() - Method in interface com.meshdynamics.api.NMS.Node
Returns the MAC-address of the parent's downlink on which this `Node` is connected.
getParentDownlinkSignal() - Method in interface com.meshdynamics.api.NMS.Node
Returns the signal RSSI in packets received by the parent's downlink interface from this node's uplink.
getParentDownlinkTxBitRate() - Method in interface com.meshdynamics.api.NMS.Node
Returns the transmit rate used by the parent for packet's transmitted to this `Node`.
getRxSignal() - Method in interface com.meshdynamics.api.NMS.ConnectedDevice
Returns the RSSI of the packets from the device to the node.
getTemperature() - Method in interface com.meshdynamics.api.NMS.Node
Returns the current node enclosure temperature.
getTreeLinkRate() - Method in interface com.meshdynamics.api.NMS.Node
Returns the 'Tree Link Rate' for the node.
getTxBitRate() - Method in interface com.meshdynamics.api.NMS.ConnectedDevice
Returns the transmit rate of packets from the node to the device.
getUnitMacAddress() - Method in interface com.meshdynamics.api.NMS.Node
Returns the MAC address of the node formatted as a string.
getUplinkSignal() - Method in interface com.meshdynamics.api.NMS.NeighborNode
Returns the RSSI of the neighbor's first downlink signal as seen by the uplink.
getUplinkSignal(int) - Method in interface com.meshdynamics.api.NMS.NeighborNode
Returns the RSSI as seen by the uplink from the specific downlink of the neighbor.
getUplinkTxBitRate() - Method in interface com.meshdynamics.api.NMS.NeighborNode

Returns the transmit rate from the uplink to the neighbor's first downlink.
getUplinkTxBitRate(int) - Method in interface com.meshdynamics.api.NMS.NeighborNode
Returns the transmit rate from the uplink to the specfic downlink of the neighbor.
getVlanConfigurationByTag(short) - Method in interface com.meshdynamics.api.NMS.Node
Returns the configuration of the specified VLAN.
getVlans() - Method in interface com.meshdynamics.api.NMS.Node
Returns an `Enumeration` of all VLANS in the `Node`.
gpsLatitude - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
Latitude coordinate of the node in decimal format.
gpsLongitude - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
Longitude coordinate of the node in decimal format.

H

heartbeatInterval - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The heartbeat interval for the node.
hexStringToBytes(String) - Static method in class com.meshdynamics.api.NMS
This utility method converts a hexadecimal string into a byte array.
hideEssid - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
When non-zero causes the ESSID field of 802.11 beacons and broadcast probe-responses to contain an empty string.
hostName - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The network host-name for the node.

I

identifier - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The identifier for the interface.
INVALID_VLAN - Static variable in class com.meshdynamics.api.NMS.ACLEntry
Constant specifying the default VLAN.
ipAddress - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The ip-address for the node in dotted decimal form.
ipAddressBytesToString(byte[]) - Static method in class com.meshdynamics.api.NMS
This utlity method converts a byte representation of IP-address to a dotted decimal dormat string.
ipAddressStringToBytes(String) - Static method in class com.meshdynamics.api.NMS
This utility method converts a dotted-decimal format string IP-address to an array of bytes.
isIpReachable() - Method in interface com.meshdynamics.api.NMS.Node
Returns non-zero if this `Node` can be communicated with using IP.
isMobile() - Method in interface com.meshdynamics.api.NMS.Node
Returns whether the node is mobile or stationary.
isRemote() - Method in interface com.meshdynamics.api.NMS.Node
Returns whether the remote or local.

K

keyIndex - Variable in class com.meshdynamics.api.NMS.WEPSecurity

The index of the key used for transmitting packets.

keys() - Method in class com.meshdynamics.api.NMS.Hashtable
Returns an `Enumeration` of all the keys in the hashtable.

L

length() - Method in class com.meshdynamics.api.NMS.ObjectArray
Retrieve the number of elements in the `ObjectArray`.

length() - Method in class com.meshdynamics.api.NMS.ShortArray
Retrieve the number of elements in the `ShortArray`.

M

macAddress - Variable in class com.meshdynamics.api.NMS.ACLEntry
The MAC-address of the device.

macAddress - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The MAC address of the interface.

macAddressBytesToHexString(byte[]) - Static method in class com.meshdynamics.api.NMS
This utility method converts a byte representation of MAC-address to a string where the individual bytes are seperated by a ':' character.

macAddressHexStringToBytes(String) - Static method in class com.meshdynamics.api.NMS
This utility method converts a string repsentation of MAC-address to an array of bytes.

manualChannel - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The channel to be used when `dynamicChannelAllocation` is set to 0.

matchCriteria - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Specifies the match criteria for the rule.

matchId - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Specifies the match identifier for the rule.

maxTransmitRate - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The maximum transmit rate for the interface.

MG_CLIENT_MODE_FORWARDER - Static variable in class com.meshdynamics.api.NMS
Specifies that the Meshdynamics Management Gateway client operates as a packet forwader, forwarding all management packets from the `Node`'s to the server.

MG_CLIENT_MODE_REMOTE_MANAGER - Static variable in class com.meshdynamics.api.NMS
Specifies that the Meshdynamics Management Gateway client operates as a remote manager, receiving management packets from remote sites.

mobilityMode - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The node's mobilty mode.

model - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The model identifier for the node.

N

name - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The name of the interface.

name - Variable in class com.meshdynamics.api.NMS.VlanConfiguration
The friendly name for the VLAN.
NETWORK_TYPE_FIPS_140_2 - Static variable in class com.meshdynamics.api.NMS
Specifies that the mesh network is a FIPS 140-2 secure network.
NETWORK_TYPE_REGULAR - Static variable in class com.meshdynamics.api.NMS
Specifies that the mesh network is a regular network.
nextSibling - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Reference to the next sibling rule object.
NMS - Class in com.meshdynamics.api
NMS is the primary class for using the **Meshdynamics Network Management System (NMS) API**.
NMS() - Constructor for class com.meshdynamics.api.NMS
Protected default constructor to be used by derived classes.
NMS.ACLConfiguration - Class in com.meshdynamics.api
Defines the Access Control List configuration for a node.
NMS.ACLConfiguration() - Constructor for class com.meshdynamics.api.NMS.ACLConfiguration
Default constructor, initializes the object with an empty entries array and sets `whiteList` to `0`.
NMS.ACLConfiguration(String) - Constructor for class com.meshdynamics.api.NMS.ACLConfiguration
Constructs the `ACLConfiguration` from a object notation string.
NMS.ACLEntry - Class in com.meshdynamics.api
Defines an Access Control List entry.
NMS.ACLEntry() - Constructor for class com.meshdynamics.api.NMS.ACLEntry
Default constructor.
NMS.ConnectedDevice - Interface in com.meshdynamics.api
Defines the properties of all devices connected to a `NMS.Node`
NMS.EffistreamRule - Class in com.meshdynamics.api
Defines a Effistream QoS rule.
NMS.EffistreamRule() - Constructor for class com.meshdynamics.api.NMS.EffistreamRule
Default constructor typically used to create the 'ROOT' object for the rules.
NMS.EffistreamRule(short, String) - Constructor for class com.meshdynamics.api.NMS.EffistreamRule
Use this constructor to create a rule without specifying child rules.
NMS.EffistreamRule(short, String, NMS.EffistreamRule) - Constructor for class com.meshdynamics.api.NMS.EffistreamRule
Use this constructor to create a rule directly specifying the first child.
NMS.EffistreamRule(short, String, short, short, short, short, short) - Constructor for class com.meshdynamics.api.NMS.EffistreamRule
Use this constructor to create a leaf-level rule object.
NMS.GeneralConfiguration - Class in com.meshdynamics.api
Defines all Node level fields used by a `NMS.Node`.
NMS.GeneralConfiguration() - Constructor for class com.meshdynamics.api.NMS.GeneralConfiguration

NMS.Hashtable - Class in com.meshdynamics.api
The Hashtable class provides an implementation of a Hashtable of generic 'Object' keys and generic 'Object' values.
NMS.Hashtable() - Constructor for class com.meshdynamics.api.NMS.Hashtable
Default constructor.
NMS.InterfaceConfiguration - Class in com.meshdynamics.api
Defines the interface level settings for a `NMS.Node`.
NMS.InterfaceConfiguration() - Constructor for class

com.meshdynamics.api.NMS.InterfaceConfiguration
Default constructor.
NMS.InterfaceConfiguration(String) - Constructor for class com.meshdynamics.api.NMS.InterfaceConfiguration
Initializes the configuration from the object notation string.
NMS.NeighborNode - Interface in com.meshdynamics.api
Defines the properties of all neighbor nodes detected by a `NMS.Node`
NMS.Network - Interface in com.meshdynamics.api
The `Network` interface defines all properties and actions associated with a mesh network.
NMS.NetworkListener - Interface in com.meshdynamics.api
The `NetworkListener` interface is used to receive events on a mesh network.
NMS.Node - Interface in com.meshdynamics.api
The `Node` interface defines all the properties and actions that can be carried out on a mesh node.
NMS.ObjectArray - Class in com.meshdynamics.api
The ObjectArray class provides an interface to a growable array that stores object references.
NMS.ObjectArray() - Constructor for class com.meshdynamics.api.NMS.ObjectArray
Default constructor to create the array with 0 elements.
NMS.ObjectArray(int) - Constructor for class com.meshdynamics.api.NMS.ObjectArray
Constructor to create the array with specified number of elements initialized to null.
NMS.ShortArray - Class in com.meshdynamics.api
Defines an array of short integers.
NMS.ShortArray(int) - Constructor for class com.meshdynamics.api.NMS.ShortArray
Constructs `ShortArray` object with specified number of elements.
NMS.ShortArray(short...) - Constructor for class com.meshdynamics.api.NMS.ShortArray
Constructs `ShortArray` object with the specified elements.
NMS.ShortArray(String) - Constructor for class com.meshdynamics.api.NMS.ShortArray
Constructs `ShortArray` object from a comma seperated list of numbers.
NMS.Thread - Class in com.meshdynamics.api
The `Thread` class provides multi-threading functionality to scripting platforms.
NMS.Thread(NMS.Thread.Runnable) - Constructor for class com.meshdynamics.api.NMS.Thread
Default constructor
NMS.Thread.Runnable - Interface in com.meshdynamics.api
The `Runnable` interface is implemented by any class whose instances are executed by a thread.
NMS.VlanConfiguration - Class in com.meshdynamics.api
Defines the settings for a Virtual-LAN in a `NMS.Node`.
NMS.VlanConfiguration() - Constructor for class com.meshdynamics.api.NMS.VlanConfiguration
Default constructor.
NMS.VlanConfiguration(String) - Constructor for class com.meshdynamics.api.NMS.VlanConfiguration
Creates a VlanConfiguration object from a object notation string.
NMS.WEPSecurity - Class in com.meshdynamics.api
Defines the information used by the IEEE 802.11 **Wired Equivalent Privacy** (WEP) setting by a Node's downlink interface.
NMS.WEPSecurity() - Constructor for class com.meshdynamics.api.NMS.WEPSecurity
Default constructor.
NMS.WPAEnterpriseSecurity - Class in com.meshdynamics.api
Defines the information used for the Wifi Protected Access security setting by a Node's downlink interface in an enterprise environment.

NMS.WPAEnterpriseSecurity() - Constructor for class com.meshdynamics.api.NMS.WPAEnterpriseSecurity
Default constructor
NMS.WPAPersonalSecurity - Class in com.meshdynamics.api
Defines the information used for the Wifi Protected Access (WPA) security setting by a node's downlink interface.
NMS.WPAPersonalSecurity() - Constructor for class com.meshdynamics.api.NMS.WPAPersonalSecurity
Default constructor
nodeDescription - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
User-defined description for the node
nodeName - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
User-defined name of the node

O

onEvent(int, NMS.Network, NMS.Node) - Method in interface com.meshdynamics.api.NMS.NetworkListener
This method is called when an event occurs on the network.
openNetwork(String, String, int) - Method in class com.meshdynamics.api.NMS
Opens the specified mesh network.
operatingChannel - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The current operating channel for the interface.
OPTION_ADHOC - Static variable in class com.meshdynamics.api.NMS
Specifies that a `Node` has the Disjoint Adhoc feature option turned on.
OPTION_ADHOC_DHCP - Static variable in class com.meshdynamics.api.NMS
Specifies that a `Node` has the DHCP server option turned on.
OPTION_ADHOC_INFRA_BEGIN - Static variable in class com.meshdynamics.api.NMS
Specifies that a `Node` has the 'Begin in infrastructure' option turned on for the Disjoint Adhoc feature.
OPTION_ADHOC_SECTORED - Static variable in class com.meshdynamics.api.NMS
Specifies that a `Node` has the 'SECTORED arbitration' option turned on for the Disjoint Adhoc feature.
OPTION_FORCED_ROOT - Static variable in class com.meshdynamics.api.NMS
Specifies that a `Node` has the Forced Root feature option turned on.
OPTION_IGMP - Static variable in class com.meshdynamics.api.NMS
Specifies that a `Node` has the IGMP multicast optimization option turned on.
OPTION_LOCATION - Static variable in class com.meshdynamics.api.NMS
Specifies that a `Node` has the 802.11 PROBE request based location tracking turned on.
OPTION_SIP - Static variable in class com.meshdynamics.api.NMS
Specifies that a `Node` has the 'SIP PHONE SYSTEM' option turned on.
options - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The combination of run-time options enabled on the node.

P

parent - Variable in class com.meshdynamics.api.NMS.EffistreamRule
Reference to the parent rule object.
PERFORMANCE_PROTOCOL_TCP - Static variable in class com.meshdynamics.api.NMS

Specifies usage of TCP protocol for running performance tests on a `Node`.

PERFORMANCE_PROTOCOL_UDP - Static variable in class com.meshdynamics.api.NMS

Specifies usage of UDP protocol for running performance tests on a `Node`.

PERFORMANCE_TYPE_DUAL_INDIVIDUAL - Static variable in class com.meshdynamics.api.NMS

Specifies that performance tests on a `Node` be run in the direction `Host -> Node` and then `Node -> Host`.

PERFORMANCE_TYPE_DUAL_SIMULTANEOUS - Static variable in class com.meshdynamics.api.NMS

Specifies that performance tests on a `Node` be run in the direction `Host -> Node` and `Node -> Host` simultaneously.

PERFORMANCE_TYPE_SINGLE - Static variable in class com.meshdynamics.api.NMS

Specifies that performance tests on a `Node` be run in the direction `Host -> Node`.

PHY_SUB_TYPE_802_11_A - Static variable in class com.meshdynamics.api.NMS

Specifies that the `InterfaceConfiguration` object contains information about a IEEE 802.11a interface.

PHY_SUB_TYPE_802_11_B - Static variable in class com.meshdynamics.api.NMS

Specifies that the `InterfaceConfiguration` object contains information about a IEEE 802.11b interface.

PHY_SUB_TYPE_802_11_BG - Static variable in class com.meshdynamics.api.NMS

Specifies that the `InterfaceConfiguration` object contains information about a mixed mode IEEE 802.11b/g interface.

PHY_SUB_TYPE_802_11_G - Static variable in class com.meshdynamics.api.NMS

Specifies that the `InterfaceConfiguration` object contains information about a IEEE 802.11g interface.

PHY_SUB_TYPE_802_11_PSF - Static variable in class com.meshdynamics.api.NMS

Specifies that the `InterfaceConfiguration` object contains information about a 20 MHz channel-width 4.9GHz interface.

PHY_SUB_TYPE_802_11_PSH - Static variable in class com.meshdynamics.api.NMS

Specifies that the `InterfaceConfiguration` object contains information about a 10 MHz channel-width 4.9GHz interface.

PHY_SUB_TYPE_802_11_PSQ - Static variable in class com.meshdynamics.api.NMS

Specifies that the `InterfaceConfiguration` object contains information about a 5 MHz channel-width 4.9GHz interface.

PHY_SUB_TYPE_IGNORE - Static variable in class com.meshdynamics.api.NMS

Specifies that the `InterfaceConfiguration` object contains information about an ETHERNET interface.

For interfaces with a `phyType` value of `PHY_TYPE_ETHERNET`, the `phySubType` shall be `PHY_SUB_TYPE_IGNORE`.

PHY_TYPE_802_11 - Static variable in class com.meshdynamics.api.NMS

Specifies that the `InterfaceConfiguration` object contains information about a IEEE 802.11 wireless interface.

PHY_TYPE_ETHERNET - Static variable in class com.meshdynamics.api.NMS

Specifies that the `InterfaceConfiguration` object contains information about an ETHERNET interface.

phySubType - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration

Defines the physical layer sub-type used by the interface.

phyType - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration

Defines the Physical layer used by the interface.

preferredParent - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration

The MAC address of the preferred parent's downlink radio.

preSharedKey - Variable in class com.meshdynamics.api.NMS.WPAPersonalSecurity

The 256-bit pre-shared key (PSK) formatted as a hexidecimal string.

put(Object, Object) - Method in class com.meshdynamics.api.NMS.Hashtable
Inserts the specified value for the specified key into the hashtable.

R

radiusServerIp - Variable in class com.meshdynamics.api.NMS.WPAEnterpriseSecurity
IP-address of the RADIUS server
radiusServerPort - Variable in class com.meshdynamics.api.NMS.WPAEnterpriseSecurity
The UDP port used by the RADIUS server
radiusServerSecret - Variable in class com.meshdynamics.api.NMS.WPAEnterpriseSecurity
The secret key used to authenticate RADIUS packets sent by the node
reboot() - Method in interface com.meshdynamics.api.NMS.Node
REBOOT's the `Node`.
rebootRequired() - Method in interface com.meshdynamics.api.NMS.Node
Returns non-zero if a 'REBOOT' is required for the `Node`.
REG_DOMAIN_CODE_CUSTOM - Static variable in class com.meshdynamics.api.NMS
Speciies the custom regulatory domain for node operation.
REG_DOMAIN_CODE_ETSI - Static variable in class com.meshdynamics.api.NMS
Specifies the ETSI regulatory domain for node operation.
REG_DOMAIN_CODE_FCC - Static variable in class com.meshdynamics.api.NMS
Specifies the FCC regulatory domain for node operation.
REG_DOMAIN_CODE_NONE - Static variable in class com.meshdynamics.api.NMS
Specifies a NULL regulatory domain for node operation.
regulatoryDomain - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration
The operating regulatory domain for the node.
remove(Object) - Method in class com.meshdynamics.api.NMS.Hashtable
Removes the specified key from the hashtable.
removeAt(int) - Method in class com.meshdynamics.api.NMS.ObjectArray
Removes the element at the specified index.
removeListener(NMS.NetworkListener) - Method in interface com.meshdynamics.api.NMS.Network
Removes the specified `NetworkListener` callback hook from the mesh network.
removeVlan(short) - Method in interface com.meshdynamics.api.NMS.Node
Removes the specified VLAN from the `Node`.
restoreDefaults() - Method in interface com.meshdynamics.api.NMS.Node
Restore's the `Node` to factory configuration.
rtsThreshold - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The 802.11 RTS threshold for the interface.
run() - Method in class com.meshdynamics.api.NMS.Thread

run() - Method in interface com.meshdynamics.api.NMS.Thread.Runnable
The `run` method implements the logic for the thread.
runPerformanceTest(int, short, short, int) - Method in interface com.meshdynamics.api.NMS.Node
Provides network performance information to the `Node`.

S

SECURITY_TYPE_NONE - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains no security parameters.

With this setting the `securityInfo` field of the `InterfaceConfiguration` is ignored and set to `null`.

SECURITY_TYPE_WEP_104 - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains security parameters for IEEE 802.11 WEP encryption using a 104-bit key.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WEPSecurity` object.

SECURITY_TYPE_WEP_40 - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains security parameters for IEEE 802.11 WEP encryption using a 40-bit key.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WEPSecurity` object.

SECURITY_TYPE_WPA2_ENTERPRISE - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access 2 encryption using a RADIUS server.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAEnterpriseSecurity` object.

SECURITY_TYPE_WPA2_PERSONAL - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access 2 encryption using a pre-shared key.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAPersonalSecurity` object.

SECURITY_TYPE_WPA_ENTERPRISE - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access encryption using a RADIUS server.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAEnterpriseSecurity` object.

SECURITY_TYPE_WPA_PERSONAL - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains security parameters for Wifi Protected Access encryption using a pre-shared key.

With this setting the `securityInfo` field of the `InterfaceConfiguration` references a `NMS.WPAPersonalSecurity` object.

securityInfo - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
Opaque object containing the security settings for the interface.

securityInfo - Variable in class com.meshdynamics.api.NMS.VlanConfiguration
Opaque object containing the security settings for the VLAN.

securityType - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The encryption/authentication scheme used to secure connections on the interface.

securityType - Variable in class com.meshdynamics.api.NMS.VlanConfiguration
The encryption/authentication scheme used to secure connections on the VLAN.

set(int, Object) - Method in class com.meshdynamics.api.NMS.ObjectArray
Set the object reference at the specified index.

set(short...) - Method in class com.meshdynamics.api.NMS.ShortArray

Set the elements of the `ShortArray` to the specified variable argument list of numbers.

set(String) - Method in class com.meshdynamics.api.NMS.ShortArray

Set the elements of the `ShortArray` from a comma seperated list of numbers.

set(int, short) - Method in class com.meshdynamics.api.NMS.ShortArray

Set the value at specified index.

setACLConfiguration(NMS.ACLConfiguration) - Method in interface com.meshdynamics.api.NMS.Node

Sets the `Node`'s Access Control List configuration.

setEffistreamRules(NMS.EffistreamRule) - Method in interface com.meshdynamics.api.NMS.Node

Updates the Effistream™ rule hierarchy for the `Node`.

setGeneralConfiguration(NMS.GeneralConfiguration) - Method in interface com.meshdynamics.api.NMS.Node

Updates the node level configuration for the `Node`.

setInterfaceConfiguration(NMS.InterfaceConfiguration) - Method in interface com.meshdynamics.api.NMS.Node

Updates the interface configuration for the `Node`.

setVlanConfiguration(NMS.VlanConfiguration) - Method in interface com.meshdynamics.api.NMS.Node

Sets the configuration of an existing VLAN in the `Node`.

setVlans(NMS.ObjectArray) - Method in interface com.meshdynamics.api.NMS.Node

Sets the `Node`'s VLAN list from a `ObjectArray`.

sleep(long) - Static method in class com.meshdynamics.api.NMS.Thread

The `sleep` method blocks the calling thread for the specified number of milli-seconds.

Since it is a static method, the calling thread does not have to be an instance of the `NMS.Thread` class.

start() - Method in class com.meshdynamics.api.NMS

Starts the node detection and event generation processes for the `NMS` object.

start() - Method in class com.meshdynamics.api.NMS.Thread

Starts the thread.

startMGClient(short, String, int, boolean, String, String, boolean) - Method in class com.meshdynamics.api.NMS

Starts the Meshdynamics Management Gateway client for remote management.

stdErrPrintln(String) - Method in class com.meshdynamics.api.NMS

Prints the specified string to the error output stream.

stdOutPrintln(String) - Method in class com.meshdynamics.api.NMS

Prints the specified string to the standard output stream.

stop() - Method in class com.meshdynamics.api.NMS

Stops the node detection and event generation processes for the `NMS` object.

stopMGClient() - Method in class com.meshdynamics.api.NMS

Stops the Meshdynamics Management Gateway client for remote management.

subnetMask - Variable in class com.meshdynamics.api.NMS.GeneralConfiguration

The subnet-mask for the node in dotted decimal form.

T

tag - Variable in class com.meshdynamics.api.NMS.VlanConfiguration

The IEEE 802.1q tag for the VLAN.

toObjectNotation() - Method in class com.meshdynamics.api.NMS.ACLConfiguration

Returns a string containing the object notation representation of the ACLConfiguration object.

toObjectNotation() - Method in class com.meshdynamics.api.NMS.ACLEntry
Returns a string containing the object notation representation of the `ACLEntry` object.
toObjectNotation() - Method in class com.meshdynamics.api.NMS.InterfaceConfiguration
Returns a string containing the object notation representation for the interface.
toObjectNotation() - Method in class com.meshdynamics.api.NMS.ObjectArray
Returns a string containing the object notation representation for the `ObjectArray`.
toObjectNotation() - Method in class com.meshdynamics.api.NMS.ShortArray
Returns a string containing the object notation representation for the `ShortArray`.
toObjectNotation() - Method in class com.meshdynamics.api.NMS.VlanConfiguration
Returns a string containing the object notation representation of the VlanConfiguration object.
toObjectNotation() - Method in class com.meshdynamics.api.NMS.WEPSecurity
Returns a string containing the object notation representation of the `WEPSecurity` object
toObjectNotation() - Method in class com.meshdynamics.api.NMS.WPAEnterpriseSecurity
Returns a string containing the object notation representation of the `WPAEnterpriseSecurity` object.
toObjectNotation() - Method in class com.meshdynamics.api.NMS.WPAPersonalSecurity
Returns a string containing the object notation representation of the `WPAPersonalSecurity` object
toString() - Method in class com.meshdynamics.api.NMS.ACLConfiguration

toString() - Method in class com.meshdynamics.api.NMS.ACLEntry

toString() - Method in class com.meshdynamics.api.NMS.EffistreamRule

toString() - Method in class com.meshdynamics.api.NMS.InterfaceConfiguration

toString() - Method in class com.meshdynamics.api.NMS.ObjectArray

toString() - Method in class com.meshdynamics.api.NMS.ShortArray

toString() - Method in class com.meshdynamics.api.NMS.VlanConfiguration

toString() - Method in class com.meshdynamics.api.NMS.WEPSecurity

toString() - Method in class com.meshdynamics.api.NMS.WPAEnterpriseSecurity

toString() - Method in class com.meshdynamics.api.NMS.WPAPersonalSecurity

toXmlSpec() - Method in class com.meshdynamics.api.NMS.EffistreamRule
Converts a EffistreamRule object hierarchy to a XML based string.
transmitPower - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
The transmit power for the interface.

U

unInitialize() - Method in class com.meshdynamics.api.NMS
Un-initializes the `NMS` instance.
unInitializeInstance() - Static method in class com.meshdynamics.api.NMS
Un-initializes the singleton instance of the `NMS` class.
upgradeFirmware(String) - Method in interface com.meshdynamics.api.NMS.Node
Upgrades the firmware of the `Node`.

USAGE_TYPE_DOWNLINK - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains information about a DOWNLINK interface.
USAGE_TYPE_SCANNER - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains information about a SCANNER interface.
USAGE_TYPE_UPLINK - Static variable in class com.meshdynamics.api.NMS
Specifies that the `InterfaceConfiguration` object contains information about an UPLINK interface.
usageType - Variable in class com.meshdynamics.api.NMS.InterfaceConfiguration
Defines the role in which the interface is used during the node's operation.

V

vlanTag - Variable in class com.meshdynamics.api.NMS.ACLEntry
The IEEE 802.1q VLAN tag to be used when the device associates.

W

waitForNodeDetect(String, long) - Method in interface com.meshdynamics.api.NMS.Network
Blocks the calling thread until all the nodes specified in `macAddresses` parameter are fully detected and configurable.
wepKeys - Variable in class com.meshdynamics.api.NMS.WEPSecurity
An array of upto 4 WEP keys formatted as hexadecimal strings.
whiteList - Variable in class com.meshdynamics.api.NMS.ACLConfiguration
Defines whether the ACL configuration entries specify a 'white-list'.

All Classes
NMS
NMS.ACLConfiguration
NMS.ACLEntry
NMS.ConnectedDevice
NMS.EffistreamRule
NMS.GeneralConfigura
NMS.Hashtable
NMS.InterfaceConfigura
NMS.NeighborNode
NMS.Network
NMS.NetworkListener
NMS.Node
NMS.ObjectArray
NMS.ShortArray
NMS.Thread
NMS.Thread.Runnable
NMS.VlanConfiguration
NMS.WEPSecurity
NMS.WPAEnterpriseSe
NMS.WPAPersonalSec

Package Class **Tree** **Deprecated** **Index** **Help**
PREV NEXT **FRAMES** **NO FRAMES**

How This API Document Is Organized

This API (Application Programming Interface) document has pages corresponding to the items in the navigation bar, described as follows.

Package

Each package has a page that contains a list of its classes and interfaces, with a summary for each. This page can contain four categories:

- Interfaces (italic)
- Classes
- Enums
- Exceptions
- Errors
- Annotation Types

Class/Interface

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three sections consisting of a class/interface description, summary tables, and detailed member descriptions:

- Class inheritance diagram
- Direct Subclasses
- All Known Subinterfaces
- All Known Implementing Classes
- Class/interface declaration
- Class/interface description

- Nested Class Summary
- Field Summary
- Constructor Summary
- Method Summary

- Field Detail
- Constructor Detail
- Method Detail

Each summary entry contains the first sentence from the detailed description for that item. The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

Annotation Type

Each annotation type has its own separate page with the following sections:

- Annotation Type declaration

- Annotation Type description
- Required Element Summary
- Optional Element Summary
- Element Detail

Enum

Each enum has its own separate page with the following sections:

- Enum declaration
- Enum description
- Enum Constant Summary
- Enum Constant Detail

Tree (Class Hierarchy)

There is a Class Hierarchy page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. The classes are organized by inheritance structure starting with `java.lang.Object`. The interfaces do not inherit from `java.lang.Object`.

- When viewing the Overview page, clicking on "Tree" displays the hierarchy for all packages.
- When viewing a particular package, class or interface page, clicking "Tree" displays the hierarchy for only that package.

Deprecated API

The Deprecated API page lists all of the API that have been deprecated. A deprecated API is not recommended for use, generally due to improvements, and a replacement API is usually given. Deprecated APIs may be removed in future implementations.

Index

The Index contains an alphabetic list of all classes, interfaces, constructors, methods, and fields.

Prev/Next

These links take you to the next or previous class, interface, package, or related page.

Frames/No Frames

These links show and hide the HTML frames. All pages are available with or without frames.

Serialized Form

Each serializable or externalizable class has a description of its serialization fields and methods. This information is of interest to re-implementors, not to developers using the API. While there is no link in the navigation bar, you can get to this information by going to any serialized class and clicking "Serialized Form" in the "See also" section of the class

description.

Constant Field Values

The Constant Field Values page lists the static final fields and their values.

This help file applies to API documentation generated using the standard doclet.

Hierarchy For All Packages

Package Hierarchies:

com.meshdynamics.api

Class Hierarchy

- java.lang.Object
  - com.meshdynamics.api.**NMS**
  - com.meshdynamics.api.**NMS.ACLConfiguration**
  - com.meshdynamics.api.**NMS.ACLEntry**
  - com.meshdynamics.api.**NMS.EffistreamRule**
  - com.meshdynamics.api.**NMS.GeneralConfiguration**
  - com.meshdynamics.api.**NMS.Hashtable**
  - com.meshdynamics.api.**NMS.InterfaceConfiguration**
  - com.meshdynamics.api.**NMS.ObjectArray**
  - com.meshdynamics.api.**NMS.ShortArray**
  - com.meshdynamics.api.**NMS.VlanConfiguration**
  - com.meshdynamics.api.**NMS.WEPSecurity**
  - com.meshdynamics.api.**NMS.WPAEnterpriseSecurity**
  - com.meshdynamics.api.**NMS.WPAPersonalSecurity**
  - java.lang.Thread (implements java.lang.Runnable)
    - com.meshdynamics.api.**NMS.Thread**

Interface Hierarchy

- com.meshdynamics.api.**NMS.ConnectedDevice**
- com.meshdynamics.api.**NMS.NeighborNode**
- com.meshdynamics.api.**NMS.Network**
- com.meshdynamics.api.**NMS.NetworkListener**
- com.meshdynamics.api.**NMS.Node**
- com.meshdynamics.api.**NMS.Thread.Runnable**