

# OpcNonExclusiveLimitAlarmNode Members

**Namespace:** Opc.UaFx

**Assemblies:** Opc.UaFx.Advanced.dll, Opc.UaFx.Advanced.dll

The [OpcNonExclusiveLimitAlarmNode](#) type exposes the following members.

## Constructors

### OpcNonExclusiveLimitAlarmNode(IOpcNode, OpcName, OpcLimitAlarmStates)

Initializes a new instance of the [OpcNonExclusiveLimitAlarmNode](#) class accessible by the **name** specified as a child node of the **parent** node given.

**C#**

```
public OpcNonExclusiveLimitAlarmNode(IOpcNode parent, OpcName name, OpcLimitAlarmStates supportedLimits)
```

#### Parameters

**parent** [IOpcNode](#)

The [IOpcNode](#) used as the parent node or a null reference (Nothing in Visual Basic) in the case there is no parent node available.

**name** [OpcName](#)

The [OpcName](#) through that the new non exclusive limit alarm node can be accessed.

**supportedLimits** [OpcLimitAlarmStates](#)

One or more members defined by the [OpcLimitAlarmStates](#) enumeration identifying the limits defined by the alarm.

### OpcNonExclusiveLimitAlarmNode(IOpcNode, OpcName, OpcNodeId, OpcLimitAlarmStates)

Initializes a new instance of the [OpcNonExclusiveLimitAlarmNode](#) class accessible by the **name** and **id** specified as a child node of the **parent** node given.

**C#**

```
public OpcNonExclusiveLimitAlarmNode(IOpcNode parent, OpcName name, OpcNodeId id, OpcLimitAlarmStates supportedLimits)
```

## Parameters

**parent** [IOpcNode](#)

The [IOpcNode](#) used as the parent node or a null reference (Nothing in Visual Basic) in the case there is no parent node available.

**name** [OpcName](#)

The [OpcName](#) through that the new non exclusive limit alarm node can be accessed.

**id** [OpcNodeId](#)

The [OpcNodeId](#) through that the new non exclusive limit alarm node can be identified and accessed.

**supportedLimits** [OpcLimitAlarmStates](#)

One or more members defined by the [OpcLimitAlarmStates](#) enumeration identifying the limits defined by the alarm.

## OpcNonExclusiveLimitAlarmNode(OpcName, OpcLimitAlarmStates)

Initializes a new instance of the [OpcNonExclusiveLimitAlarmNode](#) class accessible by the **name** specified.

**C#**

```
public OpcNonExclusiveLimitAlarmNode(OpcName name, OpcLimitAlarmStates supportedLimits)
```

## Parameters

**name** [OpcName](#)

The [OpcName](#) through that the new non exclusive limit alarm node can be accessed.

**supportedLimits** [OpcLimitAlarmStates](#)

One or more members defined by the [OpcLimitAlarmStates](#) enumeration identifying the limits defined by the alarm.

## OpcNonExclusiveLimitAlarmNode(OpcName, OpcNodeId, OpcLimitAlarmStates)

Initializes a new instance of the [OpcNonExclusiveLimitAlarmNode](#) class accessible by the **name** and **id** with the **supportedLimits** specified.

**C#**

```
public OpcNonExclusiveLimitAlarmNode(OpcName name, OpcNodeId id, OpcLimitAlarmStates supportedLimits)
```

## Parameters

**name** [OpcName](#)

The [OpcName](#) through that the new non exclusive limit alarm node can be accessed.

[id](#) [OpcNodeId](#)

The [OpcNodeId](#) through that the new non exclusive limit alarm node can be identified and accessed.

[supportedLimits](#) [OpcLimitAlarmStates](#)

One or more members defined by the [OpcLimitAlarmStates](#) enumeration identifying the limits defined by the alarm.

## Properties

### DefaultTypeDefinitionId

Gets the default identifier which identifies the node that defines the underlying node type from that this [OpcInstanceNode](#) has been created.

**C#**

```
protected override OpcNodeId DefaultTypeDefinitionId { get; }
```

#### Property Value

[OpcNodeId](#)

The [OpcNodeId](#) of the type node from that this [OpcInstanceNode](#) has been created from. These type node defines the typical structure of an instance node of its type definition. If there exists no specific type definition node a null reference (Nothing in Visual Basic).

### IsHigh

Gets or sets a value indicating whether the alarm applies to the [HighLimit](#) in a non-exclusive manner.

**C#**

```
public bool IsHigh { get; set; }
```

#### Property Value

[Boolean](#)

A value indicating the alarm circumstances using non-exclusive limits.

#### Remarks

The property is optional but at least the [IsHigh](#) or the [IsLow](#) have to be provided even though all states are optional. It is implied by the definition of a [IsHigh](#) and a [IsLow](#), that these groupings are mutually exclusive. A value cannot exceed both a [IsHigh](#) value and a [IsLow](#) value simultaneously.

# IsHighHigh

Gets or sets a value indicating whether the alarm applies to the [HighHighLimit](#) in a non-exclusive manner.

## C#

```
public bool IsHighHigh { get; set; }
```

## Property Value

[Boolean](#)

A value indicating the alarm circumstances using non-exclusive limits.

## Remarks

The property is optional but at least the [IsHigh](#) or the [IsLow](#) have to be provided even though all states are optional. It is implied by the definition of a [IsHigh](#) and a [IsLow](#), that these groupings are mutually exclusive. A value cannot exceed both a [IsHigh](#) value and a [IsLow](#) value simultaneously.

# IsHighHighNode

Gets the [OpcTwoStateVariableNode](#) of the [IsHighHigh](#) property.

## C#

```
public OpcTwoStateVariableNode IsHighHighNode { get; }
```

## Property Value

[OpcTwoStateVariableNode](#)

An instance of the [OpcTwoStateVariableNode](#) class.

# IsHighNode

Gets the [OpcTwoStateVariableNode](#) of the [IsHigh](#) property.

## C#

```
public OpcTwoStateVariableNode IsHighNode { get; }
```

## Property Value

[OpcTwoStateVariableNode](#)

An instance of the [OpcTwoStateVariableNode](#) class.

## IsLow

Gets or sets a value indicating whether the alarm applies to the [LowLimit](#) in a non-exclusive manner.

### C#

```
public bool IsLow { get; set; }
```

### Property Value

[Boolean](#)

A value indicating the alarm circumstances using non-exclusive limits.

### Remarks

The property is optional but at least the [IsHigh](#) or the [IsLow](#) have to be provided even though all states are optional. It is implied by the definition of a [IsHigh](#) and a [IsLow](#), that these groupings are mutually exclusive. A value cannot exceed both a [IsHigh](#) value and a [IsLow](#) value simultaneously.

## IsLowLow

Gets or sets a value indicating whether the alarm applies to the [LowLowLimit](#) in a non-exclusive manner.

### C#

```
public bool IsLowLow { get; set; }
```

### Property Value

[Boolean](#)

A value indicating the alarm circumstances using non-exclusive limits.

### Remarks

The property is optional but at least the [IsHigh](#) or the [IsLow](#) have to be provided even though all states are optional. It is implied by the definition of a [IsHigh](#) and a [IsLow](#), that these groupings are mutually exclusive. A value cannot exceed both a [IsHigh](#) value and a [IsLow](#) value simultaneously.

## IsLowLowNode

Gets the [OpcTwoStateVariableNode](#) of the [IsLowLow](#) property.

### C#

```
public OpcTwoStateVariableNode IsLowLowNode { get; }
```

## Property Value

[OpcTwoStateVariableNode](#)

An instance of the [OpcTwoStateVariableNode](#) class.

## IsLowNode

Gets the [OpcTwoStateVariableNode](#) of the [IsLow](#) property.

**C#**

```
public OpcTwoStateVariableNode IsLowNode { get; }
```

## Property Value

[OpcTwoStateVariableNode](#)

An instance of the [OpcTwoStateVariableNode](#) class.

## Methods

### CreateBranchCore()

Creates a new instance of the [OpcNonExclusiveLimitAlarmNode](#) using the same [Id](#) and [Name](#) as this node.

**C#**

```
protected override OpcConditionNode CreateBranchCore()
```

## Returns

[OpcConditionNode](#)

A new instance of the [OpcNonExclusiveLimitAlarmNode](#) identifiable and accessible through the same [Id](#) and [Name](#) as this node.

# Table of Contents

<b>Constructors</b>	1
OpcNonExclusiveLimitAlarmNode(IOpcNode, OpcName, OpcLimitAlarmStates)	1
OpcNonExclusiveLimitAlarmNode(IOpcNode, OpcName, OpcNodeId, OpcLimitAlarmStates)	1
OpcNonExclusiveLimitAlarmNode(OpcName, OpcLimitAlarmStates)	2
OpcNonExclusiveLimitAlarmNode(OpcName, OpcNodeId, OpcLimitAlarmStates)	2
<b>Properties</b>	3
DefaultTypeDefinitionId	3
IsHigh	3
IsHighHigh	4
IsHighHighNode	4
IsHighNode	4
IsLow	5
IsLowLow	5
IsLowLowNode	5
IsLowNode	6
<b>Methods</b>	6
CreateBranchCore()	6

