

OpcLimitAlarmNode Members

Namespace: Opc.UaFx

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

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

Constructors

OpcLimitAlarmNode(IOpcNode, OpcName, OpcLimitAlarmStates)

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

C#

```
public OpcLimitAlarmNode(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 limit alarm node can be accessed.

supportedLimits [OpcLimitAlarmStates](#)

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

OpcLimitAlarmNode(IOpcNode, OpcName, OpcNodeId, OpcLimitAlarmStates)

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

C#

```
public OpcLimitAlarmNode(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 limit alarm node can be accessed.

id [OpcNodeId](#)

The [OpcNodeId](#) through that the new 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.

OpcLimitAlarmNode(OpcName, OpcLimitAlarmStates)

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

C#

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

Parameters

name [OpcName](#)

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

supportedLimits [OpcLimitAlarmStates](#)

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

OpcLimitAlarmNode(OpcName, OpcNodeId, OpcLimitAlarmStates)

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

C#

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

Parameters

name [OpcName](#)

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

id [OpcNodeId](#)

The [OpcNodeId](#) through that the new 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).

HighHighLimit

Gets or sets a value which indicates the high high limit of a value to test for the alarm condition.

C#

```
public double HighHighLimit { get; set; }
```

Property Value

[Double](#)

The optional high high limit to test.

Remarks

The property is optional but at least one of the limit properties is required to apply limits to the alarm condition. For cases where an underlying system cannot provide the actual value of a limit, the limit property will still be provided, but will have its access level set to not readable. It is assumed that the limits are described using the same engineering unit that is assigned to the variable that is the source of the alarm. For rate of change limit alarms, it is assumed this rate is units per second unless otherwise specified.

HighHighLimitNode

Gets the [OpcPropertyNode`1](#) of the [HighHighLimit](#) property.

C#

```
public OpcPropertyNode<double> HighHighLimitNode { get; }
```

Property Value

[OpcPropertyNode<Double>](#)

An instance of the [OpcPropertyNode`1](#) class.

HighLimit

Gets or sets a value which indicates the high limit of a value to test for the alarm condition.

C#

```
public double HighLimit { get; set; }
```

Property Value

[Double](#)

The optional high limit to test.

Remarks

The property is optional but at least one of the limit properties is required to apply limits to the alarm condition. For cases where an underlying system cannot provide the actual value of a limit, the limit property will still be provided, but will have its access level set to not readable. It is assumed that the limits are described using the same engineering unit that is assigned to the variable that is the source of the alarm. For rate of change limit alarms, it is assumed this rate is units per second unless otherwise specified.

HighLimitNode

Gets the [OpcPropertyNode`1](#) of the [HighLimit](#) property.

C#

```
public OpcPropertyNode<double> HighLimitNode { get; }
```

Property Value

[OpcPropertyNode<Double>](#)

An instance of the [OpcPropertyNode`1](#) class.

LowLimit

Gets or sets a value which indicates the low limit of a value to test for the alarm condition.

C#

```
public double LowLimit { get; set; }
```

Property Value

Double

The optional low limit to test.

Remarks

The property is optional but at least one of the limit properties is required to apply limits to the alarm condition. For cases where an underlying system cannot provide the actual value of a limit, the limit property will still be provided, but will have its access level set to not readable. It is assumed that the limits are described using the same engineering unit that is assigned to the variable that is the source of the alarm. For rate of change limit alarms, it is assumed this rate is units per second unless otherwise specified.

LowLimitNode

Gets the [OpcPropertyNode`1](#) of the [LowLimit](#) property.

C#

```
public OpcPropertyNode<double> LowLimitNode { get; }
```

Property Value

OpcPropertyNode<Double>

An instance of the [OpcPropertyNode`1](#) class.

LowLowLimit

Gets or sets a value which indicates the low low limit of a value to test for the alarm condition.

C#

```
public double LowLowLimit { get; set; }
```

Property Value

Double

The optional low low limit to test.

Remarks

The property is optional but at least one of the limit properties is required to apply limits to the alarm condition. For cases where an underlying system cannot provide the actual value of a limit, the limit property will still be provided, but will have its access level set to not readable. It is assumed that the limits are described using the same engineering unit that is assigned to the variable that is the source of the alarm. For rate of change limit alarms, it is assumed this rate is units per second unless otherwise specified.

LowLowLimitNode

Gets the [OpcPropertyNode`1](#) of the [LowLowLimit](#) property.

C#

```
public OpcPropertyNode<double> LowLowLimitNode { get; }
```

Property Value

[OpcPropertyNode<Double>](#)

An instance of the [OpcPropertyNode`1](#) class.

SupportedLimits

Gets a combination of the values defined by the [OpcLimitAlarmStates](#) enumeration which defines the different limit states the [OpcLimitAlarmNode](#) supports.

C#

```
public OpcLimitAlarmStates SupportedLimits { get; }
```

Property Value

[OpcLimitAlarmStates](#)

A combination of the members defined by the [OpcLimitAlarmStates](#) enumeration.

Methods

CreateBranchCore()

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

C#

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

Returns

OpcConditionNode

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

InitializeDefaults()

Initializes the default values used by the [OpcLimitAlarmNode](#).

C#

```
protected override void InitializeDefaults()
```

Remarks

This method is used to ensure the availability of appropriate node specific default values. For more information like when this method is to be overwritten see [InitializeDefaults](#).

Table of Contents

Constructors	1
OpcLimitAlarmNode(IOpcNode, OpcName, OpcLimitAlarmStates)	1
OpcLimitAlarmNode(IOpcNode, OpcName, OpcNodeId, OpcLimitAlarmStates)	1
OpcLimitAlarmNode(OpcName, OpcLimitAlarmStates)	2
OpcLimitAlarmNode(OpcName, OpcNodeId, OpcLimitAlarmStates)	2
Properties	3
DefaultTypeDefinitionId	3
HighHighLimit	3
HighHighLimitNode	4
HighLimit	4
HighLimitNode	4
LowLimit	5
LowLimitNode	5
LowLowLimit	5
LowLowLimitNode	6
SupportedLimits	6
Methods	6
CreateBranchCore()	6
InitializeDefaults()	7