

OpcAddAnalogItemNode Members

Namespace: Opc.UaFx

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

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

Constructors

OpcAddAnalogItemNode(OpcName)

Initializes a new instance of the [OpcAddAnalogItemNode](#) class using the **name** of the analog item node to add. The according [OpcNodeId](#) to identify and access the new node is determined by the service. The new node will be a child of the [ObjectsFolder](#) node using [HasComponent](#) as the type of reference.

C#

```
public OpcAddAnalogItemNode(OpcName name)
```

Parameters

name [OpcName](#)

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

Exceptions

[ArgumentException](#)

The **name** is equals [Null](#).

[ArgumentNullException](#)

The **name** is a null reference (Nothing in Visual Basic).

OpcAddAnalogItemNode(OpcName, Object)

Initializes a new instance of the [OpcAddAnalogItemNode](#) class using the **name** of the analog item node to add. The according [OpcNodeId](#) to identify and access the new node is determined by the service. The new node will be a child of the [ObjectsFolder](#) node using [HasComponent](#) as the type of reference.

C#

```
public OpcAddAnalogItemNode(OpcName name, object value)
```

Parameters

name [OpcName](#)

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

value Object

The initial value of the new analog item node.

Exceptions

ArgumentException

The **name** is equals **Null**.

ArgumentNullException

The **name** is a null reference (Nothing in Visual Basic).

OpcAddAnalogItemNode(OpcName, OpcNodeId)

Initializes a new instance of the [OpcAddAnalogItemNode](#) class using the **name** of the analog item node to add, which shall be additionally accessible by the **nodeId** defined. The new node will be a child of the [ObjectsFolder](#) node using [HasComponent](#) as the type of reference.

C#

```
public OpcAddAnalogItemNode(OpcName name, OpcNodeId nodeId)
```

Parameters

name OpcName

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

nodeId OpcNodeId

The [OpcNodeId](#) through that the new node can be identified and accessed. In case there **Null** is specified the server will determine the according [OpcNodeId](#) by its own.

Exceptions

ArgumentException

The **name** is equals **Null**.

ArgumentNullException

The **name** or **nodeId** is a null reference (Nothing in Visual Basic).

OpcAddAnalogItemNode(OpcName, OpcNodeId, Object)

Initializes a new instance of the [OpcAddAnalogItemNode](#) class using the **name** of the analog item node to add, which shall be additionally accessible by the **nodeId** defined. The new node will be a child of the [ObjectsFolder](#) node using [HasComponent](#) as the type of reference.

C#

```
public OpcAddAnalogItemNode(OpcName name, OpcNodeId nodeId, object value)
```

Parameters

name [OpcName](#)

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

nodeId [OpcNodeId](#)

The [OpcNodeId](#) through that the new node can be identified and accessed. In case there [Null](#) is specified the server will determine the according [OpcNodeId](#) by its own.

value [Object](#)

The initial value of the new analog item node.

Exceptions

[ArgumentException](#)

The **name** is equals [Null](#).

[ArgumentNullException](#)

The **name** or **nodeId** is a null reference (Nothing in Visual Basic).

OpcAddAnalogItemNode(OpcName, OpcNodeId, OpcNodeId)

Initializes a new instance of the [OpcAddAnalogItemNode](#) class using the **name** of the analog item node to add, which shall be additionally accessible by the **nodeId** defined. The new node will be a child of the node identified by **parentNodeId** using [HasComponent](#) as the type of reference.

C#

```
public OpcAddAnalogItemNode(OpcName name, OpcNodeId nodeId, OpcNodeId parentNodeId)
```

Parameters

name [OpcName](#)

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

nodeId [OpcNodeId](#)

The [OpcNodeId](#) through that the new node can be identified and accessed. In case there [Null](#) is specified the server will determine the according [OpcNodeId](#) by its own.

parentNodeId [OpcNodeId](#)

The [OpcNodeId](#) of the parent node to reference using [HasComponent](#) as the type of reference.

Exceptions

[ArgumentException](#)

The **parentNodeId** is equals [Null](#) or **name** is equals [Null](#).

ArgumentNullException

The `name`, `nodeId` or `parentNodeId` is a null reference (Nothing in Visual Basic).

OpcAddAnalogItemNode(OpcName, OpcNodeId, OpcNodeId, Object)

Initializes a new instance of the `OpcAddAnalogItemNode` class using the `name` of the analog item node to add, which shall be additionally accessible by the `nodeId` defined. The new node will be a child of the node identified by `parentNodeId` using `HasComponent` as the type of reference.

C#

```
public OpcAddAnalogItemNode(OpcName name, OpcNodeId nodeId, OpcNodeId parentNodeId, object value)
```

Parameters

`name` `OpcName`

The `OpcName` through that the new analog item node can be accessed.

`nodeId` `OpcNodeId`

The `OpcNodeId` through that the new node can be identified and accessed. In case there `Null` is specified the server will determine the according `OpcNodeId` by its own.

`parentNodeId` `OpcNodeId`

The `OpcNodeId` of the parent node to reference using `HasComponent` as the type of reference.

`value` `Object`

The initial value of the new analog item node.

Exceptions

`ArgumentException`

The `parentNodeId` is equals `Null` or `name` is equals `Null`.

`ArgumentNullException`

The `name`, `nodeId` or `parentNodeId` is a null reference (Nothing in Visual Basic).

OpcAddAnalogItemNode(OpcName, OpcNodeId, OpcNodeId, OpcNodeId)

Initializes a new instance of the `OpcAddAnalogItemNode` class using the `name` of the analog item node to add, which shall be additionally accessible by the `nodeId` defined. The new node will be a child of the node identified by `parentNodeId` using the type of reference identified by the `referenceTypeId` specified.

C#

```
public OpcAddAnalogItemNode(OpcName name, OpcNodeId nodeId, OpcNodeId parentNodeId,
OpcNodeId referenceTypeId)
```

Parameters

name [OpcName](#)

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

nodeId [OpcNodeId](#)

The [OpcNodeId](#) through that the new node can be identified and accessed. In case there [Null](#) is specified the server will determine the according [OpcNodeId](#) by its own.

parentNodeId [OpcNodeId](#)

The [OpcNodeId](#) of the parent node to reference using the type of reference identified by [referenceTypeId](#).

referenceTypeId [OpcNodeId](#)

The [OpcNodeId](#) which identifies the type of reference to use for the new node and the existing parent node identified by [parentNodeId](#).

Exceptions

[ArgumentException](#)

The [parentNodeId](#) or [referenceTypeId](#) is equals [Null](#) or [name](#) is equals [Null](#).

[ArgumentNullException](#)

The [name](#), [nodeId](#), [parentNodeId](#) or [referenceTypeId](#) is a null reference (Nothing in Visual Basic).

OpcAddAnalogItemNode(OpcName, OpcNodeId, OpcNodeId, OpcNodeId, Object)

Initializes a new instance of the [OpcAddAnalogItemNode](#) class using the [name](#) of the analog item node to add, which shall be additionally accessible by the [nodeId](#) defined. The new node will be a child of the node identified by [parentNodeId](#) using the type of reference identified by the [referenceTypeId](#) specified.

C#

```
public OpcAddAnalogItemNode(OpcName name, OpcNodeId nodeId, OpcNodeId parentNodeId,
OpcNodeId referenceTypeId, object value)
```

Parameters

name [OpcName](#)

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

nodeId [OpcNodeId](#)

The [OpcNodeId](#) through that the new node can be identified and accessed. In case there [Null](#) is specified the server will determine the according [OpcNodeId](#) by its own.

parentNodeId **OpcNodeId**

The **OpcNodeId** of the parent node to reference using the type of reference identified by **referenceTypeId**.

referenceTypeId **OpcNodeId**

The **OpcNodeId** which identifies the type of reference to use for the new node and the existing parent node identified by **parentNodeId**.

value **Object**

The initial value of the new analog item node.

Exceptions

ArgumentException

The **parentNodeId** or **referenceTypeId** is equals **Null** or **name** is equals **Null**.

ArgumentNullException

The **name**, **nodeId**, **parentNodeId** or **referenceTypeId** is a null reference (Nothing in Visual Basic).

OpcAddAnalogItemNode(OpcName, OpcNodeId, OpcNodeId, OpcReferenceType)

Initializes a new instance of the **OpcAddAnalogItemNode** class using the **name** of the analog item node to add, which shall be additionally accessible by the **nodeId** defined. The new node will be a child of the node identified by **parentNodeId** using the **referenceType** specified as the type of reference.

C#

```
public OpcAddAnalogItemNode(OpcName name, OpcNodeId nodeId, OpcNodeId parentNodeId,
    OpcReferenceType referenceType)
```

Parameters

name **OpcName**

The **OpcName** through that the new analog item node can be accessed.

nodeId **OpcNodeId**

The **OpcNodeId** through that the new node can be identified and accessed. In case there **Null** is specified the server will determine the according **OpcNodeId** by its own.

parentNodeId **OpcNodeId**

The **OpcNodeId** of the parent node to reference using **HasComponent** as the type of reference.

referenceType **OpcReferenceType**

One of the members defined by the **OpcReferenceType** enumeration to use to setup the reference between the new node and the existing parent node identified by **parentNodeId**.

Exceptions

ArgumentException

The `parentNodeId` is equals `Null` or `name` is equals `Null`.

ArgumentNullException

The `name`, `nodeId` or `parentNodeId` is a null reference (Nothing in Visual Basic).

OpcAddAnalogItemNode(OpcName, OpcNodeId, OpcNodeId, OpcReferenceType, Object)

Initializes a new instance of the `OpcAddAnalogItemNode` class using the `name` of the analog item node to add, which shall be additionally accessible by the `nodeId` defined. The new node will be a child of the node identified by `parentNodeId` using the `referenceType` specified as the type of reference.

C#

```
public OpcAddAnalogItemNode(OpcName name, OpcNodeId nodeId, OpcNodeId parentNodeId,
    OpcReferenceType referenceType, object value)
```

Parameters

`name` `OpcName`

The `OpcName` through that the new analog item node can be accessed.

`nodeId` `OpcNodeId`

The `OpcNodeId` through that the new node can be identified and accessed. In case there `Null` is specified the server will determine the according `OpcNodeId` by its own.

`parentNodeId` `OpcNodeId`

The `OpcNodeId` of the parent node to reference using `HasComponent` as the type of reference.

`referenceType` `OpcReferenceType`

One of the members defined by the `OpcReferenceType` enumeration to use to setup the reference between the new node and the existing parent node identified by `parentNodeId`.

`value` `Object`

The initial value of the new analog item node.

Exceptions

ArgumentException

The `parentNodeId` is equals `Null` or `name` is equals `Null`.

ArgumentNullException

The `name`, `nodeId` or `parentNodeId` is a null reference (Nothing in Visual Basic).

Properties

EngineeringUnit

Gets or sets the engineering unit information which specifies the unit of the value attribute of the analog item node to add. A null reference (Nothing in Visual Basic) indicates that the property is undefined and its default value is used.

C#

```
public OpcEngineeringUnitInfo EngineeringUnit { get; set; }
```

Property Value

[OpcEngineeringUnitInfo](#)

An instance of the [OpcEngineeringUnitInfo](#) class which provides the information of the unit used to scale the value attribute of the analog item node.

EngineeringUnitRange

Gets or sets the range information which applies to normal operation. A null reference (Nothing in Visual Basic) indicates that the property is undefined and its default value is used.

C#

```
public OpcValueRange EngineeringUnitRange { get; set; }
```

Property Value

[OpcValueRange](#)

An instance of the [OpcValueRange](#) class which provides the range information of the value attribute in normal operation. This information is intended for such use as automatically scaling a bar graph display.

InstrumentRange

Gets or sets the range information which applies to the instrumental values returned by the instrument. A null reference (Nothing in Visual Basic) indicates that the property is undefined and its default value is used.

C#

```
public OpcValueRange InstrumentRange { get; set; }
```

Property Value

[OpcValueRange](#)

An instance of the [OpcValueRange](#) class which provides the range information about the values provided

by the unterlaying senor or instrument used to determine the value of the analog item node.

Table of Contents

Constructors	1
OpcAddAnalogItemNode(OpcName)	1
OpcAddAnalogItemNode(OpcName, Object)	1
OpcAddAnalogItemNode(OpcName, OpcNodeId)	2
OpcAddAnalogItemNode(OpcName, OpcNodeId, Object)	2
OpcAddAnalogItemNode(OpcName, OpcNodeId, OpcNodeId)	3
OpcAddAnalogItemNode(OpcName, OpcNodeId, OpcNodeId, Object)	4
OpcAddAnalogItemNode(OpcName, OpcNodeId, OpcNodeId, OpcNodeId)	4
OpcAddAnalogItemNode(OpcName, OpcNodeId, OpcNodeId, OpcNodeId, Object)	5
OpcAddAnalogItemNode(OpcName, OpcNodeId, OpcNodeId, OpcReferenceType)	6
OpcAddAnalogItemNode(OpcName, OpcNodeId, OpcNodeId, OpcReferenceType, Object)	7
Properties	8
EngineeringUnit	8
EngineeringUnitRange	8
InstrumentRange	8