

# OpcPropertyNode<T> Members

**Namespace:** Opc.UaFx

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

The [OpcPropertyNode<T>](#) type exposes the following members.

## Constructors

### OpcPropertyNode(IOpcNode, OpcName)

Initializes a new instance of the [OpcPropertyNode`1](#) class accessible by the [name](#) specified as a child node of the [parent](#) node given.

**C#**

```
public OpcPropertyNode(IOpcNode parent, OpcName name)
```

#### 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 property node can be accessed.

### OpcPropertyNode(IOpcNode, OpcName, OpcNodeId)

Initializes a new instance of the [OpcPropertyNode`1](#) class accessible by the [name](#) and [id](#) specified as a child node of the [parent](#) node given.

**C#**

```
public OpcPropertyNode(IOpcNode parent, OpcName name, OpcNodeId id)
```

#### 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 property node can be accessed.

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

The [OpcNodeId](#) through that the new property node can be identified and accessed.

## OpcPropertyNode(IOpcNode, OpcName, OpcNodeId, T)

Initializes a new instance of the [OpcPropertyNode`1](#) class accessible by the **name** and **id** specified with the initial value defined by **value** as a child node of the **parent** node given.

**C#**

```
public OpcPropertyNode(IOpcNode parent, OpcName name, OpcNodeId id, T value)
```

### 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 property node can be accessed.

**id** [OpcNodeId](#)

The [OpcNodeId](#) through that the new property node can be identified and accessed.

**value** [T](#)

The initial value of the new property node.

## OpcPropertyNode(IOpcNode, OpcName, T)

Initializes a new instance of the [OpcPropertyNode`1](#) class accessible by the **name** specified with the initial value defined by **value** as a child node of the **parent** node given.

**C#**

```
public OpcPropertyNode(IOpcNode parent, OpcName name, T value)
```

### 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 property node can be accessed.

**value** [T](#)

The initial value of the new property node.

## OpcPropertyNode(OpcName)

Initializes a new instance of the [OpcPropertyNode`1](#) class accessible by the **name** specified.

## C#

```
public OpcPropertyNode(OpcName name)
```

### Parameters

**name** OpcName

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

## OpcPropertyNode(OpcName, OpcNodeId)

Initializes a new instance of the **OpcPropertyNode`1** class accessible by the **name** and **id** specified.

## C#

```
public OpcPropertyNode(OpcName name, OpcNodeId id)
```

### Parameters

**name** OpcName

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

**id** OpcNodeId

The **OpcNodeId** through that the new property node can be identified and accessed.

## OpcPropertyNode(OpcName, OpcNodeId, T)

Initializes a new instance of the **OpcPropertyNode`1** class accessible by the **name** and **id** specified with the initial value given by **value**.

## C#

```
public OpcPropertyNode(OpcName name, OpcNodeId id, T value)
```

### Parameters

**name** OpcName

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

**id** OpcNodeId

The **OpcNodeId** through that the new property node can be identified and accessed.

**value** T

The initial value of the new property node.

# OpcPropertyNode(OpcName, T)

Initializes a new instance of the [OpcPropertyNode`1](#) class accessible by the [name](#) specified with the initial value given by [value](#).

**C#**

```
public OpcPropertyNode(OpcName name, T value)
```

## Parameters

[name](#) [OpcName](#)

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

[value](#) [T](#)

The initial value of the new property node.

# Properties

## DataType

Gets or sets a value which defines a pre-defined used [DataType](#) as one of the members defined by the [OpcDataType](#) enumeration to simplify querying standard data types.

**C#**

```
public override OpcDataType DataType { get; set; }
```

## Property Value

[OpcDataType](#)

One of the members defined by the [OpcDataType](#) enumeration.

# ReadPropertyGenericValueCallback

Gets or sets a callback used to read the property generic value.

**C#**

```
public OpcReadPropertyValueCallback<T> ReadPropertyGenericValueCallback { get; set; }
```

## Property Value

[OpcReadPropertyValueCallback<T>](#)

A [OpcReadPropertyValueCallback`1](#) used to read the property value as the type defined by [T](#). The value can also be a null reference (Nothing in Visual Basic).

## Remarks

This callback is used in call chain with the [ReadPropertyValueCallback](#) and will be called with the outcome of that callback routine (in case there is one defined).

# Value

Gets or sets the value of the property node.

## C#

```
public T Value { get; set; }
```

## Property Value

T

A [T](#) representing the value of the property node. This can be also a null reference (Nothing in Visual Basic).

# WritePropertyGenericValueCallback

Gets or sets a callback used to write the property generic value.

## C#

```
public OpcWritePropertyValueCallback<T> WritePropertyGenericValueCallback { get; set; }
```

## Property Value

[OpcWritePropertyValueCallback<T>](#)

A [OpcWritePropertyValueCallback`1](#) used to write the property value as the type defined by [T](#). The value can also be a null reference (Nothing in Visual Basic).

## Remarks

This callback is used in call chain with the [WritePropertyValueCallback](#) and will be called with the outcome of that callback routine (in case there is one defined).

# Methods

## InitializeDefaults()

Initializes the default values used by the [OpcPropertyNode`1](#).

## 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](#).

# ReadGenericValue(OpcReadVariableValueContext)

Reads the property node value using the `context` specified.

## C#

```
public T ReadGenericValue(OpcReadVariableValueContext context)
```

## Parameters

`context` [OpcReadVariableValueContext](#)

The [OpcReadVariableValueContext](#) to use to read the property node value.

## Returns

`T`

The `T` property node value associated with this node and read using the `context` specified. This can also be a null reference (Nothing in Visual Basic).

## Exceptions

[ArgumentNullException](#)

The `context` is a null reference (Nothing in Visual Basic).

# ReadPropertyGenericValueCore(OpcReadPropertyValueContext<OpcPropertyNode<T>>, OpcPropertyValue<T>)

Reads the value of the property node using the `context` and `value` information specified.

## C#

```
protected virtual OpcPropertyValue<T>  
ReadPropertyGenericValueCore(OpcReadPropertyValueContext<OpcPropertyNode<T>> context,  
OpcPropertyValue<T> value)
```

## Parameters

`context` [OpcReadPropertyValueContext<OpcPropertyNode`1>](#)

The [OpcReadPropertyValueContext`1](#) to use to read the property node value.

`value` [OpcPropertyValue<T>](#)

The `OpcPropertyValue` containing the currently used value constructed by the value information contained in the property node cache.

## Returns

`OpcPropertyValue<T>`

The `OpcPropertyValue` read using the `ReadPropertyGenericValueCallback` or the `value` if there is no custom callback routine defined.

# ReadPropertyValueCore(OpcReadPropertyValueContext, OpcPropertyValue<Object>)

Reads the value of the property node using the `context` and `value` information specified.

## C#

```
protected override sealed OpcPropertyValue<object>
ReadPropertyValueCore(OpcReadPropertyValueContext context, OpcPropertyValue<object> value)
```

## Parameters

`context` `OpcReadPropertyValueContext`

The `OpcReadPropertyValueContext` to use to read the property node value.

`value` `OpcPropertyValue<Object>`

The `OpcPropertyValue` containing the currently used value constructed by the value information contained in the property node cache.

## Returns

`OpcPropertyValue<Object>`

The `OpcPropertyValue` read using the `ReadPropertyValueCallback` or the `value` if there is no custom callback routine defined.

# WriteGenericValue(OpcWriteVariableValueContext, T)

Writes the `value` to the property node value using the `context` specified.

## C#

```
public void WriteGenericValue(OpcWriteVariableValueContext context, T value)
```

## Parameters

`context` `OpcWriteVariableValueContext`

The `OpcWriteVariableValueContext` to use to write the property node `value` specified.

value T

The T to write to the property node value.

## Exceptions

ArgumentNullException

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

# WritePropertyGenericValueCore(OpcWritePropertyValueContext<OpcPropertyNode<T>>, OpcPropertyValue<T>)

Writes the value of the property node using the context and value information specified.

C#

```
protected virtual OpcPropertyValue<T>
WritePropertyGenericValueCore(OpcWritePropertyValueContext<OpcPropertyNode<T>> context,
OpcPropertyValue<T> value)
```

## Parameters

context OpcWritePropertyValueContext<OpcPropertyNode`1>

The OpcWritePropertyValueContext`1 to use to write the property node value.

value OpcPropertyValue<T>

The OpcPropertyValue`1 containing the currently used value constructed by the value information contained in the property node cache.

## Returns

OpcPropertyValue<T>

The OpcPropertyValue`1 written using the WritePropertyGenericValueCallback or the value if there is no custom callback routine defined.

# WritePropertyValueCore(OpcWritePropertyValueContext, OpcPropertyValue<Object>)

Writes the value of the property node using the context and value information specified.

C#

```
protected override sealed OpcPropertyValue<object>
WritePropertyValueCore(OpcWritePropertyValueContext context, OpcPropertyValue<object> value)
```

## Parameters



## context `OpcWritePropertyValueContext`

The `OpcWritePropertyValueContext` to use to write the property node value.

## value `OpcPropertyValue<Object>`

The `OpcPropertyValue`1` containing the currently used value constructed by the value information contained in the property node cache.

## Returns

### `OpcPropertyValue<Object>`

The `OpcPropertyValue`1` written using the `WritePropertyValueCallback` or the `value` if there is no custom callback routine defined.



# Table of Contents

<b>Constructors</b>	1
OpcPropertyNode(IOpcNode, OpcName)	1
OpcPropertyNode(IOpcNode, OpcName, OpcNodeId)	1
OpcPropertyNode(IOpcNode, OpcName, OpcNodeId, T)	2
OpcPropertyNode(IOpcNode, OpcName, T)	2
OpcPropertyNode(OpcName)	2
OpcPropertyNode(OpcName, OpcNodeId)	3
OpcPropertyNode(OpcName, OpcNodeId, T)	3
OpcPropertyNode(OpcName, T)	4
<b>Properties</b>	4
DataType	4
ReadPropertyGenericValueCallback	4
Value	5
WritePropertyGenericValueCallback	5
<b>Methods</b>	5
InitializeDefaults()	5
ReadGenericValue(OpcReadVariableValueContext)	6
ReadPropertyGenericValueCore(OpcReadPropertyValueContext<OpcPropertyNode<T>>, OpcPropertyValue<T>)	6
ReadPropertyValueCore(OpcReadPropertyValueContext, OpcPropertyValue<Object>)	7
WriteGenericValue(OpcWriteVariableValueContext, T)	7
WritePropertyGenericValueCore(OpcWritePropertyValueContext<OpcPropertyNode<T>>, OpcPropertyValue<T>)	8
WritePropertyValueCore(OpcWritePropertyValueContext, OpcPropertyValue<Object>)	8