

OpcNode Members

Namespace: Opc.UaFx

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

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

Events

AfterApplyChanges

Occurs after one or more changes on the node has been notified.

C#

```
public event OpcNodeChangesEventHandler AfterApplyChanges
```

Remarks

The changes notified either belong to the [PendingChanges](#) or to changes on behalf of the node.

BeforeApplyChanges

Occurs before one or more changes on the node are notified.

C#

```
public event OpcNodeChangesEventHandler BeforeApplyChanges
```

Remarks

The changes notified either belong to the [PendingChanges](#) or to changes on behalf of the node.

Properties

Category

Gets the [NodeCategoryOpcAttribute](#) which identifies the kind of node and is therefore used to classify the node regarding its use and purpose.

C#

```
public OpcNodeCategory Category { get; }
```

Property Value

[OpcNodeCategory](#)

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

Description

Gets or sets the value of the optional [DescriptionOpcAttribute](#) which shall explain the meaning of the node.

C#

```
public OpcText Description { get; set; }
```

Property Value

[OpcText](#)

A localizable instance of the [OpcText](#) class which optionally explains the meaning of the node or a null reference (Nothing in Visual Basic) if there is no description associated with the node.

Remarks

The menaning of the node defines the usage, the context and the purpose of the node.

Descriptions

Gets the [OpcNodeGlobalization](#) instance used to control the localization and other globalization related tasks for the [Description](#) attribute of the current node.

C#

```
public OpcNodeGlobalization Descriptions { get; }
```

Property Value

[OpcNodeGlobalization](#)

An instance of the [OpcNodeGlobalization](#) class used by the current [OpcNode](#) to perform localization tasks for its [Description](#).

DisplayName

Gets or sets the value of the [DisplayNameOpcAttribute](#) which defines the localizable name of the node.

C#

```
public OpcText DisplayName { get; set; }
```

Property Value

[OpcText](#)

An instance of the [OpcText](#) class which defines the localizable name of the node.

Remarks

Client application should use this attribute if they want to display the name of the node to the user. The [String](#) part of the [DisplayName](#) is restricted to 512 characters.

DisplayNames

Gets the [OpcNodeGlobalization](#) instance used to control the localization and other globalization related tasks for the [DisplayName](#) attribute of the current node.

C#

```
public OpcNodeGlobalization DisplayNames { get; }
```

Property Value

[OpcNodeGlobalization](#)

An instance of the [OpcNodeGlobalization](#) class used by the current [OpcNode](#) to perform localization tasks for its [DisplayName](#).

HasPendingChanges

Gets a value indicating whether there exists any pending change on the node.

C#

```
public bool HasPendingChanges { get; }
```

Property Value

[Boolean](#)

The value true if [PendingChanges](#) is not equals to [None](#); otherwise the value false.

Id

Gets the value of the [NodeIdOpcAttribute](#) which unambiguously identifies the node.

C#

```
public OpcNodeId Id { get; }
```

Property Value

[OpcNodeId](#)

An instance of the [OpcNodeId](#) class representing the unambiguously identifier of the node.

Remarks

Some servers may accept alternative node identifiers in addition to the canonical node identifier represented by this attribute.

A server shall persist the node identifier of a node, that is, it shall not generate new node identifiers when rebooting.

Name

Gets or sets the value of the [BrowseNameOpcAttribute](#) which defines the non-localizable human-readable name used when browsing the address space.

C#

```
public OpcName Name { get; set; }
```

Property Value

[OpcName](#)

Remarks

The [Name](#) should never be used to display the name of the a node. Use the [DisplayName](#) instead for this purpose.

Unlike node identifiers the [Name](#) cannot be used to unambiguously identify the node. Different nodes may have the same browse name. The namespace is provided to make the browse name unique in some cases in the context of a node (e.g. properties of a node) although not unique in the context of the server. If different organizations define browse names for properties, the namespace of the [Name](#) provided by the organization makes the [Name](#) unique, although different organizations may use the same string having a slightly different meaning. Servers may often choose to use the same namespace for the node identifier and the browse name. However, if they want to provide a standard property, its browse name shall have the namespace of the standards body although the namespace of the node identifier reflects something else, for example the local server.

It is recommended that standard bodies defining standard type definitions use their namespace for the node identifier of the type definition node as well as for the browse name of the type definition node.

The [String](#)-part of the browse name is case sensitive. That is, clients shall consider them case sensitive. Server are allowed to handle browse names passed in service requests as case insensitive.

Use the value of this property to construct browse paths (see [OpcNamePath](#)).

Namespace

C#

```
public OpcNamespace Namespace { get; }
```

Property Value

OpcNamespace

Parent

Gets the parent node of the node.

C#

```
public virtual IOpcNode Parent { get; }
```

Property Value

IOpcNode

An instance implementing the [IOpcNode](#) interface which represents the physical parent of the node. In general a node can have multiple (logical) parent nodes, but this property provides the only physical parent node of the node. The value can also be a null reference (Nothing in Visual Basic) in case there the node does only have logical parent nodes.

PendingChanges

Gets a value indicating the most recent changes performed on the node since their last notification.

C#

```
public OpcNodeChanges PendingChanges { get; }
```

Property Value

OpcNodeChanges

A bitwise combination of [OpcNodeChanges](#) members identifying the changes made on the node or [None](#) if there was no change on the node since the last change notification.

Remarks

Use one of the [ApplyChanges\(OpcContext\)](#) overloads to reset the value of this property to [None](#) or use one of the [UpdateChanges\(OpcContext, OpcNodeChanges\)](#) overloads to partially reset the pending changes.

QueryEventsCallback

Gets or sets a callback used to query any event information which belongs to the node.

C#

```
public OpcQueryEventsCallback QueryEventsCallback { get; set; }
```

Property Value

[OpcQueryEventsCallback](#)

A [OpcQueryEventsCallback](#) used to query any event information which belongs to the node. The value can also be a null reference (Nothing in Visual Basic).

ReadDescriptionCallback

C#

```
public OpcReadAttributeValueCallback<OpcText> ReadDescriptionCallback { get; set; }
```

Property Value

[OpcReadAttributeValueCallback<OpcText>](#)

ReadDisplayNameCallback

C#

```
public OpcReadAttributeValueCallback<OpcText> ReadDisplayNameCallback { get; set; }
```

Property Value

[OpcReadAttributeValueCallback<OpcText>](#)

ReadUserWriteAccessCallback

C#

```
public OpcReadAttributeValueCallback<OpcAttributeWriteAccess> ReadUserWriteAccessCallback {  
get; set; }
```

Property Value

[OpcReadAttributeValueCallback<OpcAttributeWriteAccess>](#)

ReadWriteAccessCallback

C#

```
public OpcReadAttributeValueCallback<OpcAttributeWriteAccess> ReadWriteAccessCallback { get;  
set; }
```

Property Value

[OpcReadAttributeValueCallback](#)<[OpcAttributeWriteAccess](#)>

SymbolicName

C#

```
public string SymbolicName { get; set; }
```

Property Value

[String](#)

Tag

Gets or sets the object that contains additional user data about the node.

C#

```
public object Tag { get; set; }
```

Property Value

[Object](#)

An [Object](#) that contains additional user data about the node. The default is null (Nothing in Visual Basic).

UserWriteAccess

Gets or sets the value of the optional [UserWriteAccessOpcAttribute](#) which exposes the possibilities of a client to write the attributes of the node taking user access rights into account.

C#

```
public OpcAttributeWriteAccess UserWriteAccess { get; set; }
```

Property Value

[OpcAttributeWriteAccess](#)

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

Remarks

The mask value of this attribute can only further restrict the [WriteAccess](#) attribute, when it is set to not writable in the general case that applies for every user.

WriteAccess

Gets or sets the value of the optional [WriteAccessOpcAttribute](#) which exposes the possibilities of a client to write the attributes of the node.

C#

```
public OpcAttributeWriteAccess WriteAccess { get; set; }
```

Property Value

[OpcAttributeWriteAccess](#)

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

Remarks

The mask value of this attribute does not take any user access rights into account, that is, although an attribute is writable this may be restricted to a certain user / user group.

If a server does not have the ability to get the write mask information for a specific attribute from the underlying system, it should state that it is writable. If a write operation is called on the attribute, the server should transfer this request and return the corresponding [OpcStatusCode](#) if such a request is rejected.

Only in case there a corresponding attribute is set in the mask of this attribute the according attribute is writable; otherwise it can not accessed for writing.

WriteDescriptionCallback

C#

```
public OpcWriteAttributeValueCallback<OpcText> WriteDescriptionCallback { get; set; }
```

Property Value

[OpcWriteAttributeValueCallback<OpcText>](#)

WriteDisplayNameCallback

C#

```
public OpcWriteAttributeValueCallback<OpcText> WriteDisplayNameCallback { get; set; }
```

Property Value

[OpcWriteAttributeValueCallback<OpcText>](#)

WriteUserWriteAccessCallback

C#

```
public OpcWriteAttributeValueCallback<OpcAttributeWriteAccess> WriteUserWriteAccessCallback
{ get; set; }
```

Property Value

[OpcWriteAttributeValueCallback<OpcAttributeWriteAccess>](#)

WriteWriteAccessCallback

C#

```
public OpcWriteAttributeValueCallback<OpcAttributeWriteAccess> WriteWriteAccessCallback {
get; set; }
```

Property Value

[OpcWriteAttributeValueCallback<OpcAttributeWriteAccess>](#)

Methods

AddNotifier(OpcContext, IOpcNode)

C#

```
public virtual void AddNotifier(OpcContext context, IOpcNode node)
```

Parameters

context [OpcContext](#)

node [IOpcNode](#)

Exceptions

[ArgumentNullException](#)

ApplyChanges(OpcContext)

Notifies about changes performed on the node since the last notification and resets the pending changes to [None](#).

C#

```
public virtual void ApplyChanges(OpcContext context)
```

Parameters

context [OpcContext](#)

The [OpcContext](#) to use to notify about changes performed on the node.

Exceptions

[ArgumentNullException](#)

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

Remarks

This method influences the value of the properties [PendingChanges](#) and [HasPendingChanges](#). After a call to [ApplyChanges\(OpcContext\)](#) the [PendingChanges](#) property returns [None](#) and the [HasPendingChanges](#) the value false.

ApplyChanges(OpcContext, Boolean)

Notifies about changes performed on the node (and optionally on its children) since the last notification and resets the pending changes to [None](#).

C#

```
public virtual void ApplyChanges(OpcContext context, bool recursive)
```

Parameters

context [OpcContext](#)

The [OpcContext](#) to use to notify about changes performed on the node.

recursive [Boolean](#)

The value true if pending changes are to be notified for all children as well; otherwise the value false.

Exceptions

[ArgumentNullException](#)

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

Remarks

This method influences the value of the properties [PendingChanges](#) and [HasPendingChanges](#). After a call to [ApplyChanges\(OpcContext, Boolean\)](#) the [PendingChanges](#) property returns [None](#) and the [HasPendingChanges](#) the value false.

AttributeValue(OpcAttribute)

Retrieves the value of the [attribute](#) specified.

C#

```
public object AttributeValue(OpcAttribute attribute)
```

Parameters

[attribute](#) [OpcAttribute](#)

One of the members defined by the [OpcAttribute](#) enumeration which identifies the attribute its value is queried.

Returns

[Object](#)

The value of the [attribute](#) specified or a null reference (Nothing in Visual Basic) in case there the value of the attribute is a null reference or it is not supported by the current [OpcNode](#).

AttributeValue<T>(OpcAttribute)

Retrieves the value of the [attribute](#) specified.

C#

```
public T AttributeValue<T>(OpcAttribute attribute)
```

Parameters

[attribute](#) [OpcAttribute](#)

One of the members defined by the [OpcAttribute](#) enumeration which identifies the attribute its value is queried.

Returns

[T](#)

The value of the [attribute](#) as the type [T](#) specified or the default value of the type [T](#) in case there the value of the attribute is a null reference (Nothing in Visual Basic) or it is not supported by the current [OpcNode](#).

Child(OpcContext, OpcName)

Retrieves the child node its **Name** property matches exactly the **name** specified.

C#

```
public IOpcNode Child(OpcContext context, OpcName name)
```

Parameters

context OpcContext

The **OpcContext** to use to lookup the requested child node.

name OpcName

The full qualified **OpcName** of the node to lookup.

Returns

IOpcNode

An instance implementing the **IOpcNode** interface its **Name** exactly matches the **name** specified, if such a node is a known child node of this node. Otherwise a null reference (Nothing in Visual Basic).

Exceptions

ArgumentNullException

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

Children(OpcContext)

Retrieves a sequence of all nodes organized as children of this node.

C#

```
public IEnumerable<IOpcNode> Children(OpcContext context)
```

Parameters

context OpcContext

The **OpcContext** to use to lookup the child nodes.

Returns

IEnumerable<IOpcNode>

A sequence of instances implementing the **IOpcNode** interface which representing the nodes organized as child nodes of this node.

Exceptions

ArgumentNullException

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

InitializeDefaults()

Initializes the default values used by the node implementation represented / required.

C#

```
protected virtual void InitializeDefaults()
```

Remarks

This method is always called by the lowest constructor of the `OpcNode` constructor chain to initialize all default values which are explicitly necessary before the node is (for example) added as a child node of another node. In case there a parent node is passed to constructor this becomes essential to ensure that the node is initialized appropriately before it is added to the parent.

Only implement this method in case there is a custom node setup required (typically performed within the custom constructor). Any code for member field initialization or additional logic can be kept in the constructor.

IsChangePending(OpcNodeChanges)

C#

```
public bool IsChangePending(OpcNodeChanges change)
```

Parameters

`change` `OpcNodeChanges`

Returns

`Boolean`

OnAfterApplyChanges(OpcNodeChangesEventArgs)

Raises the `AfterApplyChanges` event using the event data specified.

C#

```
protected virtual void OnAfterApplyChanges(OpcNodeChangesEventArgs e)
```

Parameters

e [OpcNodeChangesEventArgs](#)

The [OpcNodeChangesEventArgs](#) with the event data.

OnBeforeApplyChanges(OpcNodeChangesEventArgs)

Raises the [BeforeApplyChanges](#) event using the event data specified.

C#

```
protected virtual void OnBeforeApplyChanges(OpcNodeChangesEventArgs e)
```

Parameters

e [OpcNodeChangesEventArgs](#)

The [OpcNodeChangesEventArgs](#) with the event data.

QueryEventsCore(OpcNodeContext, OpcEventCollection)

C#

```
protected virtual void QueryEventsCore(OpcNodeContext context, OpcEventCollection events)
```

Parameters

context [OpcNodeContext](#)

events [OpcEventCollection](#)

ReadAttributeValueCore<T>(OpcReadAttributeValueContext, OpcAttributeValue<T>)

C#

```
protected virtual OpcAttributeValue<T>  
ReadAttributeValueCore<T>(OpcReadAttributeValueContext context, OpcAttributeValue<T> value)
```

Parameters

context [OpcReadAttributeValueContext](#)

value [OpcAttributeValue<T>](#)

Returns

[OpcAttributeValue<T>](#)

RemoveNotifier(OpcContext, IOpcNode)

C#

```
public virtual void RemoveNotifier(OpcContext context, IOpcNode node)
```

Parameters

context [OpcContext](#)

node [IOpcNode](#)

Exceptions

[ArgumentNullException](#)

ReportEvent(OpcContext, OpcEvent)

C#

```
public void ReportEvent(OpcContext context, OpcEvent eventData)
```

Parameters

context [OpcContext](#)

eventData [OpcEvent](#)

Exceptions

[ArgumentNullException](#)

UpdateChanges(OpcContext, OpcNodeChanges)

Notifies about the **changes** on behalf of the node and removes pending changes which intersect with the **changes** specified.

C#

```
public void UpdateChanges(OpcContext context, OpcNodeChanges changes)
```

Parameters

context [OpcContext](#)

The [OpcContext](#) to use to notify about the **changes**.

changes [OpcNodeChanges](#)

A bitwise combination of [OpcNodeChanges](#) members to use to notify changes on behalf of the node.

Exceptions

[ArgumentNullException](#)

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

Remarks

This method influences the value of the properties [PendingChanges](#) and [HasPendingChanges](#).

UpdateChanges(OpcContext, OpcNodeChanges, Boolean)

Notifies about the **changes** on behalf of the node (and optionally on its children) and removes pending changes which intersect with the **changes** specified.

C#

```
public void UpdateChanges(OpcContext context, OpcNodeChanges changes, bool recursive)
```

Parameters

context [OpcContext](#)

The [OpcContext](#) to use to notify about changes performed on the node.

changes [OpcNodeChanges](#)

A bitwise combination of [OpcNodeChanges](#) members to use to notify changes on behalf of the node (and its children).

recursive [Boolean](#)

The value true if the **changes** are to be notified for all children as well; otherwise the value false.

Exceptions

ArgumentNullException

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

Remarks

This method influences the value of the properties `PendingChanges` and `HasPendingChanges`.

WriteAttributeValueCore<T>(OpcWriteAttributeValueContext, OpcAttributeValue<T>)

C#

```
protected virtual OpcAttributeValue<T>
WriteAttributeValueCore<T>(OpcWriteAttributeValueContext context, OpcAttributeValue<T>
value)
```

Parameters

`context` `OpcWriteAttributeValueContext`

`value` `OpcAttributeValue<T>`

Returns

`OpcAttributeValue<T>`

Table of Contents

Events	1
AfterApplyChanges	1
BeforeApplyChanges	1
Properties	1
Category	1
Description	2
Descriptions	2
DisplayName	2
DisplayNames	3
HasPendingChanges	3
Id	3
Name	4
Namespace	4
Parent	5
PendingChanges	5
QueryEventsCallback	5
ReadDescriptionCallback	6
ReadDisplayNameCallback	6
ReadUserWriteAccessCallback	6
ReadWriteAccessCallback	6
SymbolicName	7
Tag	7
UserWriteAccess	7
WriteAccess	8
WriteDescriptionCallback	8
WriteDisplayNameCallback	8
WriteUserWriteAccessCallback	9
WriteWriteAccessCallback	9
Methods	9
AddNotifier(OpcContext, IOpcNode)	9
ApplyChanges(OpcContext)	10
ApplyChanges(OpcContext, Boolean)	10
AttributeValue(OpcAttribute)	11
AttributeValue<T>(OpcAttribute)	11
Child(OpcContext, OpcName)	12
Children(OpcContext)	12
InitializeDefaults()	13
IsChangePending(OpcNodeChanges)	13
OnAfterApplyChanges(OpcNodeChangesEventArgs)	13
OnBeforeApplyChanges(OpcNodeChangesEventArgs)	14
QueryEventsCore(OpcNodeContext, OpcEventCollection)	14
ReadAttributeValueCore<T>(OpcReadAttributeValueContext, OpcAttributeValue<T>)	14
RemoveNotifier(OpcContext, IOpcNode)	15
ReportEvent(OpcContext, OpcEvent)	15
UpdateChanges(OpcContext, OpcNodeChanges)	16
UpdateChanges(OpcContext, OpcNodeChanges, Boolean)	16
WriteAttributeValueCore<T>(OpcWriteAttributeValueContext, OpcAttributeValue<T>)	17