

IOpcNode Members

Namespace: Opc.UaFx

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

The [IOpcNode](#) interface defines the following members.

Properties

Description

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

C#

```
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 meaning of the node defines the usage, the context and the purpose of the node.

DisplayName

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

C#

```
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.

HasPendingChanges

C#

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

Property Value

Boolean

Name

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

C#

```
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#

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

Property Value

[OpcNamespace](#)

Parent

C#

```
IopcNode Parent { get; }
```

Property Value

[IopcNode](#)

PendingChanges

C#

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

Property Value

[OpcNodeChanges](#)

Tag

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

C#

```
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#

```
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#

```
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.

Methods

AddNotifier(OpcContext, IOpcNode)

C#

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

Parameters

context OpcContext

node IOpcNode

Exceptions

ArgumentNullException

ApplyChanges(OpcContext)

C#

```
void ApplyChanges(OpcContext context)
```

Parameters

context OpcContext

Exceptions

ArgumentNullException

ApplyChanges(OpcContext, Boolean)

C#

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

Parameters

context OpcContext

recursive Boolean

Exceptions

[ArgumentNullException](#)

Child(OpcContext, OpcName)

Retrieves the child node its [Name](#) property matches exactly the [name](#) specified.

C#

```
IOpNode 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

[IOpNode](#)

An instance implementing the [IOpNode](#) 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#

```
IEnumerable<IOpNode> 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).

IsChangePending(OpcNodeChanges)

C#

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

Parameters

`change` [OpcNodeChanges](#)

Returns

[Boolean](#)

RemoveNotifier(OpcContext, IOpcNode)

C#

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

Parameters

`context` [OpcContext](#)

`node` [IOpcNode](#)

Exceptions

ArgumentNullException

Table of Contents

Properties	1
Description	1
DisplayName	1
HasPendingChanges	2
Name	2
Namespace	3
Parent	3
PendingChanges	3
Tag	3
UserWriteAccess	3
WriteAccess	4
Methods	4
AddNotifier(OpcContext, IOpcNode)	5
ApplyChanges(OpcContext)	5
ApplyChanges(OpcContext, Boolean)	5
Child(OpcContext, OpcName)	6
Children(OpcContext)	6
IsChangePending(OpcNodeChanges)	7
RemoveNotifier(OpcContext, IOpcNode)	7

