

OpcNodeInfo Members

Namespace: Opc.UaFx.Client

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

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

Properties

Category

Gets the classification of the node using one of the predefined categories.

C#

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

Property Value

[OpcNodeCategory](#)

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

Remarks

Using a category a node can be classified into different top level kinds of nodes like methods, objects, variables and different types.

Context

Gets the browse setup and context to use to inspect the node, its attributes and its relationships to other nodes.

C#

```
public OpcBrowseNodeContext Context { get; }
```

Property Value

[OpcBrowseNodeContext](#)

An instance of the [OpcBrowseNodeContext](#) class which has been initially created using the configuration specified by the initial browse call using the entry node.

DisplayName

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

C#

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

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.

Name

Gets the nominal information of the node represented.

C#

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

Property Value

OpcName

An instance of the [OpcName](#) class providing the nominal information about the node.

NodeId

Gets the node identifier of the node represented.

C#

```
public OpcNodeId NodeId { get; }
```

Property Value

OpcNodeId

An instance of the [OpcNodeId](#) class providing the (unresolved) node identifier of the node.

Reference

Gets the description of the node reference through that the node represented and examined has been located.

C#

```
public OpcReferenceDescription Reference { get; }
```

Property Value

OpcReferenceDescription

An instance of the [OpcReferenceDescription](#) class which provides information about the relationship through that the node has been located.

Methods

Attribute(OpcAttribute)

Retrieves the [OpcAttributeInfo](#) of the node that has the specified [OpcAttribute](#) identifier.

C#

```
public OpcAttributeInfo Attribute(OpcAttribute attribute)
```

Parameters

attribute [OpcAttribute](#)

The [OpcAttribute](#) identifier of the [OpcAttributeInfo](#) to retrieve.

Returns

[OpcAttributeInfo](#)

An [OpcAttributeInfo](#) that has the specified [OpcAttribute](#) identifier; otherwise a null reference (Nothing in Visual Basic) in case there the server does not provide such an attribute for the node.

Remarks

Although the server may provide an attribute, the [Value](#) (= an [OpcValue](#)) or its underlying raw value ([Value](#)) can be a null reference (Nothing in Visual Basic). Ensure that the [Value](#) may needs to be cast into the attribute dependent type.

The time consumed by the first call of the method depends on the availability of the server, because of the attribute is requested on-demand and is cached for subsequent calls. This means, that further attribute requests will take use of already retrieved attribute information and will not demand additional network resources.

To improve the performance to query attributes you may consider the configuration of the attributes to prefetch while browsing the nodes using [Attributes](#).

Attribute(String)

Retrieves the [OpcAttributeInfo](#) of the node that is known using the specified [browseName](#).

C#

```
public OpcAttributeInfo Attribute(string browseName)
```

Parameters

[browseName](#) [String](#)

The known browse name of the [OpcAttributeInfo](#) to retrieve.

Returns

[OpcAttributeInfo](#)

An [OpcAttributeInfo](#) that is known using the specified [browseName](#); otherwise a null reference (Nothing in Visual Basic) in case there the server does not provide such an attribute for the node.

Exceptions

[ArgumentException](#)

The [browseName](#) is an empty string.

[ArgumentNullException](#)

The [browseName](#) is a null reference (Nothing in Visual Basic).

Remarks

Although the server may provide an attribute, the [Value](#) (= an [OpcValue](#)) or its underlying raw value ([Value](#)) can be a null reference (Nothing in Visual Basic). Ensure that the [Value](#) may needs to be cast into the attribute dependent type.

The time consumed by the first call of the method depends on the availability of the server, because of the attribute is requested on-demand and is cached for subsequent calls. This means, that further attribute requests will take use of already retrieved attribute information and will not demand additional network resources.

To improve the performance to query attributes you may consider the configuration of the attributes to prefetch while browsing the nodes using [Attributes](#).

Attributes()

Retrieves for each [OpcAttribute](#) identifier the according [OpcAttributeInfo](#) of the node.

C#

```
public IEnumerable<OpcAttributeInfo> Attributes()
```

Returns

[IEnumerable<OpcAttributeInfo>](#)

A sequence of [OpcAttributeInfo](#) instances which provides the attribute identifier and attribute value of all in general defined node attributes (see [OpcAttribute](#)).

Remarks

Although the server may provide an attribute, the [Value](#) (= an [OpcValue](#)) or its underlying raw value ([Value](#)) can be a null reference (Nothing in Visual Basic). Ensure that the [Value](#) may needs to be cast into the attribute dependent type.

The time consumed by the first call of the method depends on the availability of the server, because of the attributes are requested on-demand and are cached for subsequent calls. This means, that further attribute requests will take use of already retrieved attribute information and will not demand additional network resources.

To improve the performance to query attributes you may consider the configuration of the attributes to prefetch while browsing the nodes using [Attributes](#).

Attributes(IEnumerable<OpcAttribute>)

Retrieves for each [OpcAttribute](#) identifier specified by [attributes](#) the according [OpcAttributeInfo](#) of the node.

C#

```
public IEnumerable<OpcAttributeInfo> Attributes(IEnumerable<OpcAttribute> attributes)
```

Parameters

[attributes](#) [IEnumerable<OpcAttribute>](#)

A sequence of [OpcAttribute](#) identifiers of the [OpcAttributeInfo](#) instances to retrieve.

Returns

[IEnumerable<OpcAttributeInfo>](#)

A sequence of [OpcAttributeInfo](#) instances which provides the attribute identifier and attribute value of the node attributes specified by the [OpcAttribute](#) identifiers defined by [attributes](#).

Exceptions

[ArgumentNullException](#)

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

Remarks

Although the server may provide an attribute, the `Value` (= an `OpcValue`) or its underlying raw value (`Value`) can be a null reference (Nothing in Visual Basic). Ensure that the `Value` may needs to be cast into the attribute dependent type.

The time consumed by the first call of the method depends on the availability of the server, because of the attributes are requested on-demand and are cached for subsequent calls. This means, that further attribute requests will take use of already retrieved attribute information and will not demand additional network resources.

To improve the performance to query attributes you may consider the configuration of the attributes to prefetch while browsing the nodes using `Attributes`.

Attributes(IEnumerable<String>)

Retrieves the `OpcAttributeInfo` of the node that is known using the specified `browseNames`.

C#

```
public IEnumerable<OpcAttributeInfo> Attributes(IEnumerable<string> browseNames)
```

Parameters

`browseNames` `IEnumerable<String>`

A sequence of known browse names of the `OpcAttributeInfo` to retrieve.

Returns

`IEnumerable<OpcAttributeInfo>`

A sequence of `OpcAttributeInfo` instances which provides the attribute identifier and attribute value of the node attributes specified by the `browseNames` passed.

Exceptions

`ArgumentNullException`

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

Remarks

Although the server may provide an attribute, the `Value` (= an `OpcValue`) or its underlying raw value (`Value`) can be a null reference (Nothing in Visual Basic). Ensure that the `Value` may needs to be cast into the attribute dependent type.

The time consumed by the first call of the method depends on the availability of the server, because of the attributes are requested on-demand and are cached for subsequent calls. This means, that further attribute requests will take use of already retrieved attribute information and will not demand additional network resources.

To improve the performance to query attributes you may consider the configuration of the attributes to prefetch while browsing the nodes using [Attributes](#).

Attributes(OpcAttribute[])

Retrieves for each [OpcAttribute](#) identifier specified by [attributes](#) the according [OpcAttributeInfo](#) of the node.

C#

```
public IEnumerable<OpcAttributeInfo> Attributes(params OpcAttribute[] attributes)
```

Parameters

[attributes](#) [OpcAttribute\[\]](#)

An array of [OpcAttribute](#) identifiers of the [OpcAttributeInfo](#) instances to retrieve.

Returns

[IEnumerable<OpcAttributeInfo>](#)

An array of [OpcAttributeInfo](#) instances which provides the attribute identifier and attribute value of the node attributes specified by the [OpcAttribute](#) identifiers defined by [attributes](#).

Exceptions

[ArgumentNullException](#)

The [attributes](#) array is a null reference (Nothing in Visual Basic).

Remarks

Although the server may provide an attribute, the [Value](#) (= an [OpcValue](#)) or its underlying raw value ([Value](#)) can be a null reference (Nothing in Visual Basic). Ensure that the [Value](#) may needs to be cast into the attribute dependent type.

The time consumed by the first call of the method depends on the availability of the server, because of the attributes are requested on-demand and are cached for subsequent calls. This means, that further attribute requests will take use of already retrieved attribute information and will not demand additional network resources.

To improve the performance to query attributes you may consider the configuration of the attributes to prefetch while browsing the nodes using [Attributes](#).

Attributes(String[])

Retrieves the [OpcAttributeInfo](#) of the node that is known using the specified [browseNames](#).

C#

```
public IEnumerable<OpcAttributeInfo> Attributes(params string[] browseNames)
```

Parameters

[browseNames](#) [String\[\]](#)

An array of known browse names of the [OpcAttributeInfo](#) to retrieve.

Returns

[IEnumerable<OpcAttributeInfo>](#)

A sequence of [OpcAttributeInfo](#) instances which provides the attribute identifier and attribute value of the node attributes specified by the [browseNames](#) passed.

Exceptions

[ArgumentNullException](#)

The [browseNames](#) array is a null reference (Nothing in Visual Basic).

Remarks

Although the server may provide an attribute, the [Value](#) (= an [OpcValue](#)) or its underlying raw value ([Value](#)) can be a null reference (Nothing in Visual Basic). Ensure that the [Value](#) may needs to be cast into the attribute dependent type.

The time consumed by the first call of the method depends on the availability of the server, because of the attributes are requested on-demand and are cached for subsequent calls. This means, that further attribute requests will take use of already retrieved attribute information and will not demand additional network resources.

To improve the performance to query attributes you may consider the configuration of the attributes to prefetch while browsing the nodes using [Attributes](#).

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 [OpcNodeInfo](#).

Remarks

The time consumed by the first call of the method depends on the availability of the server, because of the attribute is requested on-demand and is cached for subsequent calls. This means, that further attribute requests will take use of already retrieved attribute information and will not demand additional network resources.

To improve the performance to query attributes you may consider the configuration of the attributes to prefetch while browsing the nodes using [Attributes](#).

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 [OpcNodeInfo](#).

Remarks

The time consumed by the first call of the method depends on the availability of the server, because of the attribute is requested on-demand and is cached for subsequent calls. This means, that further attribute

requests will take use of already retrieved attribute information and will not demand additional network resources.

To improve the performance to query attributes you may consider the configuration of the attributes to prefetch while browsing the nodes using [Attributes](#).

Child(String)

Retrieves the [OpcNodeInfo](#) of the child node that is known using the specified **name** of the node.

C#

```
public OpcNodeInfo Child(string name)
```

Parameters

name [String](#)

The nominal information of the child node to retrieve.

Returns

[OpcNodeInfo](#)

An [OpcNodeInfo](#) that is known using the specified **name**; otherwise a null reference (Nothing in Visual Basic) in case there the server does not provide such a child node for the node.

Exceptions

[ArgumentNullException](#)

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

Remarks

The time consumed by the first call of the method depends on the availability of the server, because of the child is requested on-demand and is cached for subsequent calls. This means, that further child requests will take use of already retrieved child information and will not demand additional network resources.

Children()

Retrieves for each child node of the node represented the according [OpcNodeInfo](#).

C#

```
public IEnumerable<OpcNodeInfo> Children()
```

Returns

IEnumerable<OpcNodeInfo>

A sequence of [OpcNodeInfo](#) instances which provides the node information of all nodes referenced as child (see [OpcNodeInfo](#)).

Remarks

The time consumed by the first call of the method depends on the availability of the server, because of the nodes are requested on-demand and are cached for subsequent calls. This means, that further child requests will take use of already retrieved child information and will not demand additional network resources.

Children(OpcNodeCategory)

Retrieves for each child node of the [category](#) specified of the node represented the according [OpcNodeInfo](#).

C#

```
public IEnumerable<OpcNodeInfo> Children(OpcNodeCategory category)
```

Parameters

[category](#) [OpcNodeCategory](#)

One of the members defined by the [OpcNodeCategory](#) enumeration which defines the classification of the child nodes to retrieve or a combination of the members.

Returns

[IEnumerable<OpcNodeInfo>](#)

A sequence of [OpcNodeInfo](#) instances which provides the node information of all nodes referenced as child of the [category](#) specified.

Remarks

The time consumed by the first call of the method depends on the availability of the server, because of the nodes are requested on-demand and are cached for subsequent calls. This means, that further child requests will take use of already retrieved child information and will not demand additional network resources.

Children(OpcReferenceType)

Retrieves for each child node referenced using the [referenceType](#) specified the according [OpcNodeInfo](#).

C#

```
public IEnumerable<OpcNodeInfo> Children(OpcReferenceType referenceType)
```

Parameters

referenceType [OpcReferenceType](#)

One of the members defined by the [OpcReferenceType](#) enumeration which defines the type of reference between the node represented and child nodes to retrieve.

Returns

[IEnumerable](#)<[OpcNodeInfo](#)>

A sequence of [OpcNodeInfo](#) instances which provides the node information of all nodes referenced as child using the **referenceType** specified.

Remarks

The time consumed by the first call of the method depends on the availability of the server, because of the nodes are requested on-demand and are cached for subsequent calls. This means, that further child requests will take use of already retrieved child information and will not demand additional network resources.

In case there the **referenceType** specified is not part of the reference types browsed using the [Context](#) of the current [OpcNodeInfo](#) (see [ReferenceTypeIds](#)) there is separate browse operation performed to query the children requested. In this case the child nodes are not cached and are re-queried on every call to this method.

Parent(String)

Retrieves the [OpcNodeInfo](#) of the parent node that is known using the specified **name** of the node.

C#

```
public OpcNodeInfo Parent(string name)
```

Parameters

name [String](#)

The nominal information of the parent node to retrieve.

Returns

[OpcNodeInfo](#)

An [OpcNodeInfo](#) that is known using the specified **name**; otherwise a null reference (Nothing in Visual Basic) in case there the server does not provide such a parent node for the node.

Exceptions

[ArgumentNullException](#)

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

Remarks

The time consumed by the first call of the method depends on the availability of the server, because of the parent is requested on-demand and is cached for subsequent calls. This means, that further parent requests will take use of already retrieved parent information and will not demand additional network resources.

Parents()

Retrieves for each parent node of the node represented the according [OpcNodeInfo](#).

C#

```
public IEnumerable<OpcNodeInfo> Parents()
```

Returns

[IEnumerable<OpcNodeInfo>](#)

A sequence of [OpcNodeInfo](#) instances which provides the node information of all nodes referenced as parent (see [OpcNodeInfo](#)).

Remarks

The time consumed by the first call of the method depends on the availability of the server, because of the nodes are requested on-demand and are cached for subsequent calls. This means, that further parent requests will take use of already retrieved parent information and will not demand additional network resources.

Parents(OpcNodeCategory)

Retrieves for each parent node of the `category` specified of the node represented the according [OpcNodeInfo](#).

C#

```
public IEnumerable<OpcNodeInfo> Parents(OpcNodeCategory category)
```

Parameters

`category` [OpcNodeCategory](#)

One of the members defined by the [OpcNodeCategory](#) enumeration which defines the classification of the parent nodes to retrieve or a combination of the members.

Returns

[IEnumerable<OpcNodeInfo>](#)

A sequence of [OpcNodeInfo](#) instances which provides the node information of all nodes referenced as parent of the [category](#) specified.

Remarks

The time consumed by the first call of the method depends on the availability of the server, because of the nodes are requested on-demand and are cached for subsequent calls. This means, that further parent requests will take use of already retrieved parent information and will not demand additional network resources.

Parents(OpcReferenceType)

Retrieves for each parent node referenced using the [referenceType](#) specified the according [OpcNodeInfo](#).

C#

```
public IEnumerable<OpcNodeInfo> Parents(OpcReferenceType referenceType)
```

Parameters

[referenceType](#) [OpcReferenceType](#)

One of the members defined by the [OpcReferenceType](#) enumeration which defines the type of reference between the node represented and parent nodes to retrieve.

Returns

[IEnumerable<OpcNodeInfo>](#)

A sequence of [OpcNodeInfo](#) instances which provides the node information of all nodes referenced as parent using the [referenceType](#) specified.

Remarks

The time consumed by the first call of the method depends on the availability of the server, because of the nodes are requested on-demand and are cached for subsequent calls. This means, that further child requests will take use of already retrieved child information and will not demand additional network resources.

In case there the [referenceType](#) specified is not part of the reference types browsed using the [Context](#) of the current [OpcNodeInfo](#) (see [ReferenceTypeIds](#)) there is separate browse operation performed to query the parents requested. In this case the parent nodes are not cached and are re-queried on every call to this method.

Table of Contents

Properties	1
Category	1
Context	1
DisplayName	1
Name	2
NodeId	2
Reference	2
Methods	3
Attribute(OpcAttribute)	3
Attribute(String)	4
Attributes()	4
Attributes(IEnumerable<OpcAttribute>)	5
Attributes(IEnumerable<String>)	6
Attributes(OpcAttribute[])	7
Attributes(String[])	8
AttributeValue(OpcAttribute)	8
AttributeValue<T>(OpcAttribute)	9
Child(String)	10
Children()	10
Children(OpcNodeCategory)	11
Children(OpcReferenceType)	11
Parent(String)	12
Parents()	13
Parents(OpcNodeCategory)	13
Parents(OpcReferenceType)	14

