OpcVariableNodeInfo Members

Namespace: Opc.UaFx.Client
Assemblies: Opc.UaFx.Advanced.dll, Opc.UaFx.Advanced.dll
The OpcVariableNodeInfo type exposes the following members.

Properties

AccessLevel

Gets the value of the AccessLevel attribute of the variable node represented.

C#

```
public OpcAccessLevel AccessLevel { get; }
```

Property Value

OpcAccessLevel

One of the members defined by the OpcAccessLevel enumeration. For more information about the use of the AccessLevel attribute see AccessLevel and OpcAccessLevel.

Remarks

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

ArrayDimensions

Gets the length for each dimension of an array value of the variable node represented.

C#

```
public uint[] ArrayDimensions { get; }
```

Property Value

UInt32[]

A null reference (Nothing in Visual Basic) in case there the value of the variable node represented is a scalar value; otherwise an UInt32 array there each entry indicates a single dimension while the value of the entry defines the length of that dimension.

Remarks

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

ArrayLength

Gets the total number of elements in all the dimensions of the array value of the variable node represented.

C#

```
public long ArrayLength { get; }
```

Property Value

Int64

The total number of elements in all the dimensions of the array value of the variable node represented; zero if there are no elements in the array.

Remarks

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

ArrayRank

Gets the rank (number of dimensions) of the array value of the variable node represented. For example, a one-dimensional array returns 1, a two-dimensional array returns 2, and so on.

C#

```
public int ArrayRank { get; }
```

Property Value

Int32

The rank (number of dimensions) of the array value of the variable node represented.

Remarks

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

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

DataType

Gets the OpcTypeNodeInfo about the data type node of the variable node represented.

C#

```
public OpcTypeNodeInfo DataType { get; }
```

Property Value

OpcTypeNodeInfo

An instance of the OpcTypeNodeInfo representing the type definition of the variable node represented. In case there is no known type definition for the data type referenced by the variable node a null reference (Nothing in Visual Basic).

DataTypeId

Gets the OpcNodeId about the data type node of the variable node represented.

C#

```
public OpcNodeId DataTypeId { get; }
```

Property Value

OpcNodeId

An instance of the OpcNodeId which identifies the type node definition of the variable node represented. In case there is no type definition for the type of value provided by the variable node represented a null reference (Nothing in Visual Basic).

Remarks

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

IsArray

Gets a value indicating whether the value of the variable represented is an array.

C#

```
public bool IsArray { get; }
```

Property Value

Boolean

The value true if the value of the variable represented is an array; otherwise the false.

Remarks

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

IsHistorizing

Gets a value indicating whether the server is actively collecting data for the history of the variable node represented.

C#

```
public bool IsHistorizing { get; }
```

Property Value

Boolean

The value true if the server is actively collecting historical data; otherwise the value false.

Remarks

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

UserAccessLevel

Gets the value of the UserAccessLevel attribute of the variable node represented.

C#

```
public OpcAccessLevel UserAccessLevel { get; }
```

Property Value

OpcAccessLevel

One of the members defined by the OpcAccessLevel enumeration. For more information about the use of the UserAccessLevel attribute see AccessLevel and OpcAccessLevel.

Remarks

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

UserWriteAccess

Gets a mask that indicates which attributes are writable by the current user of the client of a node taking user access rights into account.

C#

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

Property Value

OpcAttributeWriteAccess

A logical combination of the members defined by the OpcAttributeWriteAccess enumeration indicating which OpcAttributes of the variable node represented are currently writable by the current user of the client.

Remarks

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

WriteAccess

Gets a mask that indicates which attributes are generally writable through the server not taking user access rights into account.

C#

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

Property Value

OpcAttributeWriteAccess

A logical combination of the members defined by the OpcAttributeWriteAccess enumeration indicating which OpcAttributes of the variable node represented are generally writable by the server.

Remarks

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

Methods

GetArrayLength(Int32)

Gets a 32-bit integer that represents the number of elements in the specified dimension of the array value of the variable node.

C#

```
public long GetArrayLength(int dimension)
```

Parameters

`dimension` Int32

A zero-based dimension of the array value whose length needs to be determined.

Returns

Int64

A 32-bit integer that represents the number of elements in the specified dimension of the array value.

Exceptions

ArgumentOutOfRangeException

The `dimension` is less than zero or equals to or greater than ArrayRank.

InvalidOperationException

The value of the variable node represented is not an array (see IsArray).

Remarks

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

GetArrayLowerBound(Int32)

Gets the index of the first element of the specified dimension in the array value of the variable node.

C#

```
public long GetArrayLowerBound(int dimension)
```

Parameters

`dimension` Int32

A zero-based dimension of the array value whose starting index needs to be determined.

Returns

Int64

The index of the first element of the specified dimension in the array value.

Exceptions

ArgumentOutOfRangeException

The `dimension` is less than zero or equals to or greater than ArrayRank.

InvalidOperationException

The value of the variable node represented is not an array (see IsArray).

GetArrayUpperBound(Int32)

Gets the index of the last element of the specified dimension in the array value of the variable node.

C#

```
public long GetArrayUpperBound(int dimension)
```

Parameters

`dimension` Int32

A zero-based dimension of the array value whose upper bound needs to be determined.

Returns

Int64

The index of the last element of the specified dimension in the array value, or -1 if the specified dimension is empty.

Exceptions

ArgumentOutOfRangeException

The `dimension` is less than zero or equals to or greater than ArrayRank.

InvalidOperationException

The value of the variable node represented is not an array (see IsArray).

Remarks

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

Table of Contents

Properties 1
AccessLevel 1
ArrayDimensions 1
ArrayLength 2
ArrayRank 2
DataType 3
DataTypeId 3
IsArray 3
IsHistorizing 4
UserAccessLevel 4
UserWriteAccess 5
WriteAccess 5
Methods 6
GetArrayLength(Int32) 6
GetArrayLowerBound(Int32) 6
GetArrayUpperBound(Int32) 7