

IOpcDataTypeInfo Members

Namespace: Opc.UaFx

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

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

Properties

BaseType

Gets the type from which the current [IOpcDataTypeInfo](#) directly inherits.

C#

```
IOpcDataTypeInfo BaseType { get; }
```

Property Value

[IOpcDataTypeInfo](#)

An instance of the [IOpcDataTypeInfo](#) which represents the type from which the current [IOpcDataTypeInfo](#) directly inherits or a null reference (Nothing in Visual Basic) if the type does not inherit from a different [IOpcDataTypeInfo](#).

Encoding

Gets the information used to identify the type of encoding used for the data of the type represented.

C#

```
OpcEncoding Encoding { get; }
```

Property Value

[OpcEncoding](#)

An instance of the [OpcEncoding](#) class which represents the information used to encode data of the type or a null reference (Nothing in Visual Basic) if is no encoding information associated with the type.

EncodingMask

Gets the characteristics used to encode the existence of optional fields.

C#

```
OpcEncodingMask EncodingMask { get; }
```

Property Value

OpcEncodingMask

An instance of the [OpcEncodingMask](#) class.

Encodings

Gets information of all encodings supported by the type represented.

C#

```
OpcEncoding[] Encodings { get; }
```

Property Value

[OpcEncoding\[\]](#)

An array of [OpcEncoding](#) instances supported by the type.

IsArray

Gets a value indicating whether the type is an array.

C#

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

Property Value

[Boolean](#)

The value true if the current type is an array; otherwise the value false.

IsEnum

Gets a value indicating whether the current [IOpcDataTypeInfo](#) represents an enumeration.

C#

```
bool IsEnum { get; }
```

Property Value

[Boolean](#)

The value true if the current [IOpcDataTypeInfo](#) represents an enumeration; otherwise the value false.

Remarks

The OPC UA handles enumerations as a numeric value that has a fixed set of valid values. The encoded value described by an enumeration is always an unsigned integer (see [UInt32](#)) with a fixed size.

IsOpaque

Gets a value indicating whether the current [IOpcDataTypeInfo](#) describes a primitive fixed size type.

C#

```
bool IsOpaque { get; }
```

Property Value

Boolean

The value true if the current [IOpcDataTypeInfo](#) represents an opaque type; otherwise the value false.

IsPrimitive

Gets a value indicating whether the [IOpcDataTypeInfo](#) is one of the primitive types.

C#

```
bool IsPrimitive { get; }
```

Property Value

Boolean

The value true if the [IOpcDataTypeInfo](#) is one of the primitive types; otherwise the value false.

IsStruct

Gets a value indicating whether the [IOpcDataTypeInfo](#) is a structured type; that is not a value type.

C#

```
bool IsStruct { get; }
```

Property Value

Boolean

The value true if the [IOpcDataTypeInfo](#) is a structured type; otherwise the value false.

Remarks

The OPC UA handles structures as a sequence of values. Each value in the sequence is called a field. A field may specify that zero, one or multiple instances of the type appear within the sequence described by the structure. Some fields have lengths that are not multiples of 8 bits. Several of these fields may appear in a sequence in a structure, however, the total number of bits used in the sequence shall be aligned on a byte boundary. A sequence of fields which do not line up on byte boundaries are specified from the least significant bit to the most significant bit. Sequences which are longer than one byte overflow from the

most significant bit to the first byte into the least significant bit of the next byte.

IsSystemType

Gets a value indicating whether the current [IOpcDataTypeInfo](#) represents one of the predefined built-in data types defined by the OPC UA.

C#

```
bool IsSystemType { get; }
```

Property Value

Boolean

The value true if the current data type represents a OPC UA is a built-in type; otherwise the value false.

Name

Gets a value which defines the non-localizable human-readable name of the type represented. A node which represents this [IOpcDataTypeInfo](#) uses the [Name](#) as its [BrowseName](#).

C#

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

Property Value

OpcName

An instance of the [OpcName](#) class with the [String](#) used as the name of the type which does not unambiguously identify the [IOpcDataTypeInfo](#).

Size

Gets the size of the type represented as the number of bytes required.

C#

```
long? Size { get; }
```

Property Value

Nullable<Int64>

The size of the type as the number of bytes required; if this [IOpcDataTypeInfo](#) represents a type with a fixed size (see [IsOpaque](#)).

SizeInBits

Gets the size of the type represented as the number of bits required.

C#

```
long? SizeInBits { get; }
```

Property Value

[Nullable<Int64>](#)

The size of the type as the number of bits required; if this [IOpcDataTypeInfo](#) represents a type with a fixed size (see [IsOpaque](#)).

TypeId

Gets a value which identifies the type represented. A node which represents this [IOpcDataTypeInfo](#) uses the [TypeId](#) as its [NodeId](#).

C#

```
OpcNodeId TypeId { get; }
```

Property Value

[OpcNodeId](#)

An instance of the [OpcNodeId](#) class used as the identifier of the type which unambiguously identifies the [IOpcDataTypeInfo](#).

UnderlyingType

Gets the type which provides the implementation of the [IOpcDataTypeInfo](#).

C#

```
Type UnderlyingType { get; }
```

Property Value

[Type](#)

The [Type](#) which implements the type declared or a null reference (Nothing in Visual Basic) if there does not exist a declaration of the type described by this [IOpcDataTypeInfo](#).

Methods

GetArrayRank()

Gets the number of dimensions in an array.

C#

```
int GetArrayRank()
```

Returns

[Int32](#)

An integer that contains the number of dimensions in the current type.

Exceptions

[ArgumentException](#)

The current type is not an array.

GetElementType()

Returns the [IOpcDataTypeInfo](#) of the object encompassed or referred to by the current array.

C#

```
IOpcDataTypeInfo GetElementType()
```

Returns

[IOpcDataTypeInfo](#)

The [IOpcDataTypeInfo](#) of the object encompassed or referred to by the current array or null if the current [IOpcDataTypeInfo](#) is not an array.

GetField(String)

Searches for the field with the specified `name`.

C#

```
IOpcDataFieldInfo GetField(string name)
```

Parameters

`name` [String](#)

The [String](#) containing the name of the data field to get.

Returns

IOpcDataFieldInfo

An instance of the [IOpcDataFieldInfo](#) representing the field with the specified name, if found; otherwise a null reference (Nothing in Visual Basic).

Exceptions

[ArgumentNullException](#)

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

Remarks

The `name` specified is compared using [Ordinal](#).

GetFields()

Returns all the fields of the current [IOpcDataTypeInfo](#).

C#

```
IOpcDataFieldInfo[] GetFields()
```

Returns

[IOpcDataFieldInfo](#)[]

An array of [IOpcDataFieldInfo](#) objects representing all the fields defined for the current [IOpcDataTypeInfo](#) or an empty array of type [IOpcDataFieldInfo](#) if no fields are defined for the current [IOpcDataTypeInfo](#).

MakeArrayType()

Returns a [IOpcDataTypeInfo](#) object representing a one-dimensional array of the current type, with a lower bound of zero.

C#

```
IOpcDataTypeInfo MakeArrayType()
```

Returns

[IOpcDataTypeInfo](#)

A [IOpcDataTypeInfo](#) object representing a one-dimensional array of the current type, with a lower bound of zero.

MakeArrayType(Int32)

Returns a [IOpcDataTypeInfo](#) object representing an array of the current type, with the specified number of dimensions.

C#

```
IopcDataTypeInfo MakeArrayType(int rank)
```

Parameters

rank [Int32](#)

The number of dimensions for the array. This number must be less than or equal to 32.

Returns

[IopcDataTypeInfo](#)

An object representing an array of the current type, with the specified number of dimensions.

Exceptions

[ArgumentOutOfRangeException](#)

The **rank** is not between 1 and 32 (inclusive).

Table of Contents

Properties	1
BaseType	1
Encoding	1
EncodingMask	1
Encodings	2
IsArray	2
IsEnum	2
IsOpaque	3
IsPrimitive	3
IsStruct	3
IsSystemType	4
Name	4
Size	4
SizeInBits	5
TypeId	5
UnderlyingType	5
Methods	5
GetArrayRank()	6
GetElementType()	6
GetField(String)	6
GetFields()	7
MakeArrayType()	7
MakeArrayType(Int32)	7

