

OpcDataTypeDictionary Members

Namespace: Opc.UaFx

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

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

Properties

ByteOrder

Gets the default byte order of the data types defined in this [OpcDataTypeDictionary](#).

C#

```
public virtual OpcByteOrder? ByteOrder { get; }
```

Property Value

[Nullable<OpcByteOrder>](#)

One of the members defined by the [OpcByteOrder](#) enumeration or a null reference (Nothing in Visual Basic) in case there the dictionary does not declare a specific byte order to use.

Documentation

Gets any semantic information that would help a human to understand what the type dictionary defines.

C#

```
public virtual string Documentation { get; }
```

Property Value

[String](#)

A human readable text that describes the type dictionary and that would help a human to understand what the type dictionary defines.

Empty

Gets the type dictionary to use if there is no specific type dictionary data available.

C#

```
public static OpcDataTypeDictionary Empty { get; }
```

Property Value

OpcDataTypeDictionary

An instance of the [OpcDataTypeDictionary](#) class which refers to the [Empty](#).

EmptyType

Gets the type to use if there is no specific type data available, but the empty type shall refer to the current [OpcDataTypeDictionary](#).

C#

```
public OpcDataTypeInfo EmptyType { get; }
```

Property Value

[OpcDataTypeInfo](#)

An instance of the [OpcDataTypeInfo](#) class which refers to the current [OpcDataTypeDictionary](#).

EncodingType

Gets the type of encoding used for the data types defined in the current [OpcDataTypeDictionary](#).

C#

```
public OpcEncodingType EncodingType { get; }
```

Property Value

[OpcEncodingType](#)

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

HasCachedTypes

Gets a value indicating whether the type dictionary already has determined at least some type information offered as [OpcDataTypeInfo](#) objects.

C#

```
protected bool HasCachedTypes { get; }
```

Property Value

[Boolean](#)

The value true if the dictionary offers already prepared [OpcDataTypeInfo](#) objects; otherwise the value false.

HasCachingCompleted

Gets a value indicating whether the type dictionary has determined the whole type information offered as [OpcDataTypeInfo](#) objects.

C#

```
protected virtual bool HasCachingCompleted { get; }
```

Property Value

[Boolean](#)

The value true if the dictionary has completed its type information retrieval and will use its internal cache to determine [OpcDataTypeInfo](#) objects; otherwise the value false.

IsEmpty

Gets a value indicating whether the current [OpcDataTypeDictionary](#) represents a type dictionary to use if there is no specific type dictionary described.

C#

```
public virtual bool IsEmpty { get; }
```

Property Value

[Boolean](#)

The value true if the type dictionary does not declare a specific type dictionary; otherwise the value false.

Name

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

C#

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

Property Value

[OpcName](#)

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

Namespace

Gets the namespace within the type dictionary has been declared.

C#

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

Property Value

OpcNamespace

An instance of the [OpcNamespace](#) class which provides the information about the namespace within the current [OpcDataTypeDictionary](#) defines its type data.

NodeId

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

C#

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

Property Value

OpcNodeId

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

TypeSystem

Gets the the [OpcDataTypeSystem](#) which contains the data dictionary represented.

C#

```
public OpcDataTypeSystem TypeSystem { get; }
```

Property Value

OpcDataTypeSystem

An instance of the [OpcDataTypeSystem](#) class which defines the current [OpcDataTypeDictionary](#).

XmlNamespace

Gets the [Namespace](#) used in the XML-based declaration of the type dictionary.

C#

```
public virtual string XmlNamespace { get; }
```

Property Value

String

A [String](#) which represents the [Namespace](#) used in the XML-based declaration of the type dictionary.

Methods

GetType(OpcEncoding)

Retrieves the [OpcDataTypeInfo](#) object which declares the type which is identified by the [encoding](#) specified.

C#

```
public OpcDataTypeInfo GetType(OpcEncoding encoding)
```

Parameters

[encoding](#) [OpcEncoding](#)

The [OpcEncoding](#) which identifies the [OpcDataTypeInfo](#) to retrieve.

Returns

[OpcDataTypeInfo](#)

The [OpcDataTypeInfo](#) object which is known under the [encoding](#) specified or a null reference (Nothing in Visual Basic) if there doesn't exist a known [OpcDataTypeInfo](#) which is associated with the [encoding](#) specified.

Exceptions

[ArgumentNullException](#)

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

GetType(OpcName)

Retrieves the [OpcDataTypeInfo](#) object which is known under the [name](#) specified.

C#

```
public OpcDataTypeInfo GetType(OpcName name)
```

Parameters

[name](#) [OpcName](#)

The [OpcName](#) of the [OpcDataTypeInfo](#) object to retrieve.

Returns

OpcDataTypeInfo

The [OpcDataTypeInfo](#) object its [Name](#) is equals to the [name](#) specified; otherwise a null reference (Nothing in Visual Basic).

Exceptions

ArgumentNullException

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

GetType(OpcNodeId)

Retrieves the [OpcDataTypeInfo](#) object which declares the type which is identified by the [typeId](#) specified.

C#

```
public OpcDataTypeInfo GetType(OpcNodeId typeId)
```

Parameters

[typeId](#) [OpcNodeId](#)

The [OpcNodeId](#) which identifies the [OpcDataTypeInfo](#) to retrieve.

Returns

OpcDataTypeInfo

The [OpcDataTypeInfo](#) object which is known under the [typeId](#) specified or a null reference (Nothing in Visual Basic) if there doesn't exist a known [OpcDataTypeInfo](#) object which is associated with the [typeId](#) specified.

Exceptions

ArgumentNullException

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

GetType(String)

Retrieves the [OpcDataTypeInfo](#) object which is known under the [name](#) specified.

C#

```
public OpcDataTypeInfo GetType(string name)
```

Parameters

name String

The String to use to identify the [OpcDataTypeInfo](#) object to retrieve.

Returns

[OpcDataTypeInfo](#)

The [OpcDataTypeInfo](#) object its **Name** is equals (regarding its **Value**) to the **name** specified; otherwise a null reference (Nothing in Visual Basic).

Exceptions

[ArgumentException](#)

The **name** is equals [Empty](#).

[ArgumentNullException](#)

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

GetType(Type)

Retrieves the [OpcDataTypeInfo](#) object which declares the type implemented by the **underlyingType** specified.

C#

```
public OpcDataTypeInfo GetType(Type underlyingType)
```

Parameters

underlyingType Type

The Type which implements the [OpcDataTypeInfo](#) to retrieve.

Returns

[OpcDataTypeInfo](#)

The [OpcDataTypeInfo](#) object which declares the **underlyingType** specified or a null reference (Nothing in Visual Basic) if there isn't a [OpcDataTypeInfo](#) object associated with the **underlyingType** specified.

Exceptions

[ArgumentNullException](#)

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

GetType(XmlQualifiedName)

Retrieves the [OpcDataTypeInfo](#) object which is known under the [xmlName](#) specified.

C#

```
public OpcDataTypeInfo GetType(XmlQualifiedName xmlName)
```

Parameters

[xmlName](#) [XmlQualifiedName](#)

The [XmlQualifiedName](#) of the [OpcDataTypeInfo](#) object to retrieve.

Returns

[OpcDataTypeInfo](#)

The [OpcDataTypeInfo](#) object its [XmlName](#) is equals to the [xmlName](#) specified; otherwise a null reference (Nothing in Visual Basic).

Exceptions

[ArgumentNullException](#)

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

GetTypeCore(OpcEncoding)

Retrieves the [OpcDataTypeInfo](#) object which declares the type which is identified by the [encoding](#) specified.

C#

```
protected virtual OpcDataTypeInfo GetTypeCore(OpcEncoding encoding)
```

Parameters

[encoding](#) [OpcEncoding](#)

The [OpcEncoding](#) which identifies the [OpcDataTypeInfo](#) to retrieve.

Returns

[OpcDataTypeInfo](#)

The [OpcDataTypeInfo](#) object which is known under the [encoding](#) specified or a null reference (Nothing in Visual Basic) if there doesn't exist a known [OpcDataTypeInfo](#) which is associated with the [encoding](#) specified.

Remarks

It is already assured that the passed **encoding** is not a null reference (Nothing in Visual Basic). In case there this method is not implemented in a derived class this method determines the type from a distinct and on-demand determined sequence which combines the **OpcDataTypeInfo** objects offered by **RetrieveTypesCore** and **RetrieveTypesCore(IDictionary)**. Both methods are called only once and their output is cached for subsequent type information retrieval.

GetTypeCore(OpcName)

Retrieves the **OpcDataTypeInfo** object which is known under the **name** specified.

C#

```
protected virtual OpcDataTypeInfo GetTypeCore(OpcName name)
```

Parameters

name **OpcName**

The **OpcName** of the **OpcDataTypeInfo** object to retrieve.

Returns

OpcDataTypeInfo

The **OpcDataTypeInfo** object its **Name** is equals to the **name** specified; otherwise a null reference (Nothing in Visual Basic).

Remarks

It is already assured that the passed **name** is not a null reference (Nothing in Visual Basic). In case there this method is not implemented in a derived class this method determines the type from a distinct and on-demand determined sequence which combines the **OpcDataTypeInfo** objects offered by **RetrieveTypesCore** and **RetrieveTypesCore(IDictionary)**. Both methods are called only once and their output is cached for subsequent type information retrieval.

GetTypeCore(OpcNodeId)

Retrieves the **OpcDataTypeInfo** object which declares the type which is identified by the **typeId** specified.

C#

```
protected virtual OpcDataTypeInfo GetTypeCore(OpcNodeId typeId)
```

Parameters

typeId **OpcNodeId**

The **OpcNodeId** which identifies the **OpcDataTypeInfo** to retrieve.

Returns

OpcDataTypeInfo

The [OpcDataTypeInfo](#) object which is known under the [typeId](#) specified or a null reference (Nothing in Visual Basic) if there doesn't exist a known [OpcDataTypeInfo](#) object which is associated with the [typeId](#) specified.

Remarks

It is already assured that the passed [typeId](#) is not a null reference (Nothing in Visual Basic). In case there this method is not implemented in a derived class this method determines the type from a distinct and on-demand determined sequence which combines the [OpcDataTypeInfo](#) objects offered by [RetrieveTypesCore](#) and [RetrieveTypesCore\(IDictionary\)](#). Both methods are called only once and their output is cached for subsequent type information retrieval.

GetTypeCore(String)

Retrieves the [OpcDataTypeInfo](#) object which is known under the [name](#) specified.

C#

```
protected virtual OpcDataTypeInfo GetTypeCore(string name)
```

Parameters

[name](#) String

The [String](#) to use to identify the [OpcDataTypeInfo](#) object to retrieve.

Returns

[OpcDataTypeInfo](#)

The [OpcDataTypeInfo](#) object its [Name](#) is equals (regarding its [Value](#)) to the [name](#) specified; otherwise a null reference (Nothing in Visual Basic).

Remarks

It is already assured that the passed [name](#) is not a null reference (Nothing in Visual Basic) nor equals [Empty](#). In case there this method is not implemented in a derived class this method determines the type from a distinct and on-demand determined sequence which combines the [OpcDataTypeInfo](#) objects offered by [RetrieveTypesCore](#) and [RetrieveTypesCore\(IDictionary\)](#). Both methods are called only once and their output is cached for subsequent type information retrieval.

GetTypeCore(Type)

Retrieves the [OpcDataTypeInfo](#) object which declares the type implemented by the [underlyingType](#) specified.

C#

```
protected virtual OpcDataTypeInfo GetTypeCore(Type underlyingType)
```

Parameters

underlyingType Type

The **Type** which implements the **OpcDataTypeInfo** to retrieve.

Returns

OpcDataTypeInfo

The **OpcDataTypeInfo** object which declares the **underlyingType** specified or a null reference (Nothing in Visual Basic) if there isn't a **OpcDataTypeInfo** object associated with the **underlyingType** specified.

Remarks

It is already assured that the passed **underlyingType** is not a null reference (Nothing in Visual Basic). In case there this method is not implemented in a derived class this method determines the type from a distinct and on-demand determined sequence which combines the **OpcDataTypeInfo** objects offered by **RetrieveTypesCore** and **RetrieveTypesCore(IDictionary)**. Both methods are called only once and their output is cached for subsequent type information retrieval.

GetTypeCore(XmlQualifiedName)

Retrieves the **OpcDataTypeInfo** object which is known under the **xmlName** specified.

C#

```
protected virtual OpcDataTypeInfo GetTypeCore(XmlQualifiedName xmlName)
```

Parameters

xmlName XmlQualifiedName

The **XmlQualifiedName** of the **OpcDataTypeInfo** object to retrieve.

Returns

OpcDataTypeInfo

The **OpcDataTypeInfo** object its **XmlName** is equals to the **xmlName** specified; otherwise a null reference (Nothing in Visual Basic).

Remarks

It is already assured that the passed **xmlName** is not a null reference (Nothing in Visual Basic). In case there this method is not implemented in a derived class this method determines the type from a distinct and on-demand determined sequence which combines the **OpcDataTypeInfo** objects offered by **RetrieveTypesCore** and **RetrieveTypesCore(IDictionary)**. Both methods are called only once and their output is cached for subsequent type information retrieval.

GetTypes()

Retrieves all [OpcDataTypeInfo](#) objects offered by the [OpcDataTypeDictionary](#).

C#

```
public OpcDataTypeInfo[] GetTypes()
```

Returns

[OpcDataTypeInfo\[\]](#)

An array that contains all [OpcDataTypeInfo](#) objects that are offered by the [OpcDataTypeDictionary](#).

GetXmlNamespace(OpcNamespace)

Determines the namespace information used to initialize [XmlQualifiedName](#) instances to identify [OpcDataTypeInfo](#) objects using a full qualified name.

C#

```
public static string GetXmlNamespace(OpcNamespace namespace)
```

Parameters

namespace [OpcNamespace](#)

The [OpcNamespace](#) to use to determine the namespace information used for [XmlQualifiedName](#) instances.

Returns

[String](#)

A [String](#) representing the used [XmlQualifiedName.Namespace](#) in the context of the **namespace** specified.

RetrieveTypesCore()

Retrieves all [OpcDataTypeInfo](#) objects explicitly offered by the [OpcDataTypeDictionary](#).

C#

```
protected IDictionary<XmlQualifiedName, OpcDataTypeInfo> RetrieveTypesCore()
```

Returns

[IDictionary<XmlQualifiedName, OpcDataTypeInfo>](#)

A dictionary of explicitly offered [OpcDataTypeInfo](#) objects.

RetrieveTypesCore(IDictionary<XmlQualifiedName, OpcDataTypeInfo>)

When implemented in a derived class, retrieves all [OpcDataTypeInfo](#) objects on-demand offered by the [OpcDataTypeDictionary](#) except the already [knownTypes](#) specified.

C#

```
protected virtual IEnumerable<OpcDataTypeInfo>
RetrieveTypesCore(IDictionary<XmlQualifiedName, OpcDataTypeInfo> knownTypes)
```

Parameters

[knownTypes](#) [IDictionary<XmlQualifiedName, OpcDataTypeInfo>](#)

The dictionary of already known [OpcDataTypeInfo](#) objects which shall not be offered.

Returns

[IEnumerable<OpcDataTypeInfo>](#)

A sequence of [OpcDataTypeInfo](#) objects offered by the [OpcDataTypeDictionary](#) except the types in [knownTypes](#).

ToString()

Returns a [String](#) representing the [Name](#) and the [Namespace](#) of the current [OpcDataTypeDictionary](#).

C#

```
public override string ToString()
```

Returns

[String](#)

A [String](#) representing the [Name](#) and the [Namespace](#) of the current [OpcDataTypeDictionary](#).

Table of Contents

Properties	1
ByteOrder	1
Documentation	1
Empty	1
EmptyType	2
EncodingType	2
HasCachedTypes	2
HasCachingCompleted	3
IsEmpty	3
Name	3
Namespace	3
NodeId	4
TypeSystem	4
XmlNamespace	4
Methods	5
GetType(OpcEncoding)	5
GetType(OpcName)	5
GetType(OpcNodeId)	6
GetType(String)	6
GetType(Type)	7
GetType(XmlQualifiedName)	8
GetTypeCore(OpcEncoding)	8
GetTypeCore(OpcName)	9
GetTypeCore(OpcNodeId)	9
GetTypeCore(String)	10
GetTypeCore(Type)	10
GetTypeCore(XmlQualifiedName)	11
GetTypes()	12
GetXmlNamespace(OpcNamespace)	12
RetrieveTypesCore()	12
RetrieveTypesCore(IDictionary<XmlQualifiedName, OpcDataTypeInfo>)	13
ToString()	13