

OpcDataTypeSystem Members

Namespace: Opc.UaFx

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

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

Properties

Binary

Gets the type system used for [Binary](#) encoded type information offered by the Client or Server application.

C#

```
public static OpcDataTypeSystem Binary { get; }
```

Property Value

[OpcDataTypeSystem](#)

An instance of the [OpcDataTypeSystem](#) class which contains the type dictionaries described by the [Types](#) using the [OpcDataTypeEncodingAttribute](#) with the [OpcEncodingTypeBinary](#) implemented in the application which uses the OPC UA SDK.

Default

Gets the default type system used by the OPC UA stack.

C#

```
public static OpcDataTypeSystem Default { get; }
```

Property Value

[OpcDataTypeSystem](#)

An instance of the [OpcDataTypeSystem](#) class which contains the type dictionaries used by default in Client and Server applications.

Remarks

The information offered by the default [OpcDataTypeSystem](#) is embedded in the OPC UA SDK and is therefore determined from the resources of the assembly whenever required.

Empty

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

C#

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

Property Value

[OpcDataTypeSystem](#)

An instance of the [OpcDataTypeSystem](#) class which contains no type dictionary nor type information.

EmptyDictionary

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

C#

```
public OpcDataTypeDictionary EmptyDictionary { get; }
```

Property Value

[OpcDataTypeDictionary](#)

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

EncodingType

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

C#

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

Property Value

[OpcEncodingType](#)

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

HasCachedDictionaries

Gets a value indicating whether the type system already has determined at least some type dictionaries offered as [OpcDataTypeDictionary](#) objects.

C#

```
protected bool HasCachedDictionaries { get; }
```

Property Value

Boolean

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

HasCachingCompleted

Gets a value indicating whether the type system has determined all type dictionaries offered as [OpcDataTypeDictionary](#) objects.

C#

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

Property Value

Boolean

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

Name

Gets a value which defines the human-readable name of the type system represented. A node which represents this [OpcDataTypeSystem](#) 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 system which does not unambiguously identify the [OpcDataTypeSystem](#).

NodeId

Gets a value which identifies the type system represented. A node which represents this [OpcDataTypeSystem](#) 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 system which unambiguously identifies the [OpcDataTypeSystem](#).

Xml

Gets the type system used for [Xml](#) encoded type information offered by the Client or Server application.

C#

```
public static OpcDataTypeSystem Xml { get; }
```

Property Value

[OpcDataTypeSystem](#)

An instance of the [OpcDataTypeSystem](#) class which contains the type dictionaries described by the [Types](#) using the [OpcDataTypeEncodingAttribute](#) with the [OpcEncodingTypeXml](#) implemented in the application which uses the OPC UA SDK.

Methods

GetDictionaries()

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

C#

```
public OpcDataTypeDictionary[] GetDictionaries()
```

Returns

[OpcDataTypeDictionary](#)[]

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

GetDictionary(OpcName)

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

C#

```
public OpcDataTypeDictionary GetDictionary(OpcName name)
```

Parameters

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

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

Returns

OpcDataTypeDictionary

The [OpcDataTypeDictionary](#) 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).

GetDictionary(OpcNamespace)

Retrieves the [OpcDataTypeDictionary](#) object which is known under the [namespace](#) specified.

C#

```
public OpcDataTypeDictionary GetDictionary(OpcNamespace namespace)
```

Parameters

[namespace](#) [OpcNamespace](#)

The [OpcNamespace](#) of the [OpcDataTypeDictionary](#) object to retrieve.

Returns

OpcDataTypeDictionary

The [OpcDataTypeDictionary](#) object its [Namespace](#) is equals to the [namespace](#) specified; otherwise a null reference (Nothing in Visual Basic).

Exceptions

ArgumentNullException

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

GetDictionary(OpcNodeId)

Retrieves the [OpcDataTypeDictionary](#) object which represents the type dictionary which is identified by the [nodeId](#) specified.

C#

```
public OpcDataTypeDictionary GetDictionary(OpcNodeId nodeId)
```

Parameters

nodeId OpcNodeId

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

Returns

[OpcDataTypeDictionary](#)

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

Exceptions

[ArgumentNullException](#)

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

GetDictionary(String)

Retrieves the [OpcDataTypeDictionary](#) object which is known under the [nameOrNamespace](#) specified.

C#

```
public OpcDataTypeDictionary GetDictionary(string nameOrNamespace)
```

Parameters

[nameOrNamespace](#) String

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

Returns

[OpcDataTypeDictionary](#)

The [OpcDataTypeDictionary](#) object its [Name](#) (regarding its [Value](#)) or its [Namespace](#) (regarding its [Value](#)) or its [XmlNamespace](#) is equals to the [nameOrNamespace](#) specified; otherwise a null reference (Nothing in Visual Basic).

Exceptions

[ArgumentException](#)

The [nameOrNamespace](#) is equals [Empty](#).

[ArgumentNullException](#)

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

GetDictionaryCore(OpcName)

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

C#

```
protected virtual OpcDataTypeDictionary GetDictionaryCore(OpcName name)
```

Parameters

name [OpcName](#)

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

Returns

[OpcDataTypeDictionary](#)

The [OpcDataTypeDictionary](#) 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 dictionary from a distinct and on-demand determined sequence which combines the [OpcDataTypeDictionary](#) objects offered by [RetrieveDictionariesCore](#) and [RetrieveDictionariesCore\(ICollection\)](#). Both methods are called only once and their output is cached for subsequent type dictionary retrieval.

GetDictionaryCore(OpcNamespace)

Retrieves the [OpcDataTypeDictionary](#) object which is known under the **namespace** specified.

C#

```
protected virtual OpcDataTypeDictionary GetDictionaryCore(OpcNamespace namespace)
```

Parameters

namespace [OpcNamespace](#)

The [OpcNamespace](#) of the [OpcDataTypeDictionary](#) object to retrieve.

Returns

[OpcDataTypeDictionary](#)

The [OpcDataTypeDictionary](#) object its [Namespace](#) is equals to the **namespace** specified; otherwise a null reference (Nothing in Visual Basic).

Remarks

It is already assured that the passed `namespace` 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 dictionary from a distinct and on-demand determined sequence which combines the `OpcDataTypeDictionary` objects offered by `RetrieveDictionariesCore` and `RetrieveDictionariesCore(IList)`. Both methods are called only once and their output is cached for subsequent type dictionary retrieval.

GetDictionaryCore(OpcNodeId)

Retrieves the `OpcDataTypeDictionary` object which represents the type dictionary which is identified by the `nodeId` specified.

C#

```
protected virtual OpcDataTypeDictionary GetDictionaryCore(OpcNodeId nodeId)
```

Parameters

`nodeId` `OpcNodeId`

The `OpcNodeId` which identifies the `OpcDataTypeDictionary` to retrieve.

Returns

`OpcDataTypeDictionary`

The `OpcDataTypeDictionary` object which is known under the `nodeId` specified or a null reference (Nothing in Visual Basic) if there doesn't exist a known `OpcDataTypeDictionary` object which is associated with the `nodeId` specified.

Remarks

It is already assured that the passed `nodeId` 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 dictionary from a distinct and on-demand determined sequence which combines the `OpcDataTypeDictionary` objects offered by `RetrieveDictionariesCore` and `RetrieveDictionariesCore(IList)`. Both methods are called only once and their output is cached for subsequent type dictionary retrieval.

GetDictionaryCore(String)

Retrieves the `OpcDataTypeDictionary` object which is known under the `nameOrNamespace` specified.

C#

```
protected virtual OpcDataTypeDictionary GetDictionaryCore(string nameOrNamespace)
```

Parameters

`nameOrNamespace` `String`

The `String` to use to identify the `OpcDataTypeDictionary` object to retrieve.

Returns

OpcDataTypeDictionary

The [OpcDataTypeDictionary](#) object its [Name](#) (regarding its [Value](#)) or its [Namespace](#) (regarding its [Value](#)) or its [XmlNamespace](#) is equals to the [nameOrNamespace](#) specified; otherwise a null reference (Nothing in Visual Basic).

Remarks

It is already assured that the passed [nameOrNamespace](#) 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 dictionary from a distinct and on-demand determined sequence which combines the [OpcDataTypeDictionary](#) objects offered by [RetrieveDictionariesCore](#) and [RetrieveDictionariesCore\(IList\)](#). Both methods are called only once and their output is cached for subsequent type dictionary retrieval.

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(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 determined from the sequence of dictionaries offered by [RetrieveDictionariesCore](#) and [RetrieveDictionariesCore\(IList\)](#). Both

methods are called only once and their output is cached for subsequent type dictionary retrieval. These dictionaries are then queried for the type information requested.

GetTypeCore(OpcName)

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

C#

```
protected override 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 determined from the sequence of dictionaries offered by [RetrieveDictionariesCore](#) and [RetrieveDictionariesCore\(IList\)](#). Both methods are called only once and their output is cached for subsequent type dictionary retrieval. These dictionaries are then queried for the type information requested.

GetTypeCore(OpcNodeId)

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

C#

```
protected override 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 determined from the sequence of dictionaries offered by [RetrieveDictionariesCore](#) and [RetrieveDictionariesCore\(IList\)](#). Both methods are called only once and their output is cached for subsequent type dictionary retrieval. These dictionaries are then queried for the type information requested.

GetTypeCore(String)

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

C#

```
protected override 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 determined from the sequence of dictionaries offered by [RetrieveDictionariesCore](#) and [RetrieveDictionariesCore\(IList\)](#). Both methods are called only once and their output is cached for subsequent type dictionary retrieval. These dictionaries are then queried for the type information requested.

GetTypeCore(Type)

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

C#

```
protected override 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 determined from the sequence of dictionaries offered by **RetrieveDictionariesCore** and **RetrieveDictionariesCore(IList)**. Both methods are called only once and their output is cached for subsequent type dictionary retrieval. These dictionaries are then queried for the type information requested.

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 determined from the sequence of dictionaries offered by **RetrieveDictionariesCore** and **RetrieveDictionariesCore(IList)**. Both

methods are called only once and their output is cached for subsequent type dictionary retrieval. These dictionaries are then queried for the type information requested.

GetTypesCore()

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

C#

```
protected override OpcDataTypeInfo[] GetTypesCore()
```

Returns

[OpcDataTypeInfo\[\]](#)

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

Of(OpcEncodingType)

Retrieves the type system used for the [encodingType](#) specified which provides encoded type information offered by the Client or Server application.

C#

```
public static OpcDataTypeSystem Of(OpcEncodingType encodingType)
```

Parameters

[encodingType](#) [OpcEncodingType](#)

One of the members defined by the [OpcEncodingType](#) enumeration which identifies the type system to retrieve.

Returns

[OpcDataTypeSystem](#)

An instance of the [OpcDataTypeSystem](#) class which contains the type dictionaries described by the [Types](#) using the [OpcDataTypeEncodingAttribute](#) with the [OpcEncodingType](#) specified by [encodingType](#) and implemented in the application which uses the OPC UA SDK.

RetrieveDictionariesCore()

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

C#

```
protected IList<OpcDataTypeDictionary> RetrieveDictionariesCore()
```

Returns

IList<OpcDataTypeDictionary>

A list of explicitly offered [OpcDataTypeDictionary](#) objects.

RetrieveDictionariesCore(IList<OpcDataTypeDictionary>)

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

C#

```
protected virtual IEnumerable<OpcDataTypeDictionary>
RetrieveDictionariesCore(IList<OpcDataTypeDictionary> knownDictionaries)
```

Parameters

[knownDictionaries](#) [IList<OpcDataTypeDictionary>](#)

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

Returns

[IEnumerable<OpcDataTypeDictionary>](#)

A sequence of [OpcDataTypeDictionary](#) objects offered by the [OpcDataTypeSystem](#) except the type dictionaries in [knownDictionaries](#).

ToString()

Returns a [String](#) representing the [Name](#) of the current [OpcDataTypeSystem](#).

C#

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

Returns

[String](#)

A [String](#) representing the [Name](#) of the current [OpcDataTypeSystem](#).

Table of Contents

Properties	1
Binary	1
Default	1
Empty	2
EmptyDictionary	2
EncodingType	2
HasCachedDictionaries	2
HasCachingCompleted	3
Name	3
NodeId	3
Xml	4
Methods	4
GetDictionaries()	4
GetDictionary(OpcName)	4
GetDictionary(OpcNamespace)	5
GetDictionary(OpcNodeId)	5
GetDictionary(String)	6
GetDictionaryCore(OpcName)	7
GetDictionaryCore(OpcNamespace)	7
GetDictionaryCore(OpcNodeId)	8
GetDictionaryCore(String)	8
GetType(OpcEncoding)	9
GetType(XmlQualifiedName)	9
GetTypeCore(OpcEncoding)	10
GetTypeCore(OpcName)	11
GetTypeCore(OpcNodeId)	11
GetTypeCore(String)	12
GetTypeCore(Type)	12
GetTypeCore(XmlQualifiedName)	13
GetTypesCore()	14
Of(OpcEncodingType)	14
RetrieveDictionariesCore()	14
RetrieveDictionariesCore(IList<OpcDataTypeDictionary>)	15
ToString()	15