OpcData Members

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

Constructors

OpcData(IOpcDataStore)

Initializes a new instance of the OpcData class using the `dataStore` specified.

C#

```
protected OpcData(IOpcDataStore dataStore)
```

Parameters

`dataStore` IOpcDataStore

The IOpcDataStore of the object data to represent.

Exceptions

ArgumentNullException

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

Events

TypeResolve

Occurs when the resolution of a data type fails. Use this event to manually provide the Type to use for a specific data type definition identifier.

C#

```
public static event OpcTypeResolveEventHandler TypeResolve
```

Properties

DataStore

Gets the IOpcDataStore which contains the data of this OpcData instance.

C#

```
protected IOpcDataStore DataStore { get; }
```

Property Value

IOpcDataStore

An instance implementing the IOpcDataStore interface containing the object data represented by this OpcData instance.

Methods

DeepCopy()

C#

```
protected virtual object DeepCopy()
```

Returns

Object

GetType(OpcNodeId)

Retrieves the Type which implementes the data type which is identified by the `typeId` specified.

C#

```
public static Type GetType(OpcNodeId typeId)
```

Parameters

`typeId` OpcNodeId

The OpcNodeId which identifies the Type to retrieve.

Returns

Type

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

Exceptions

ArgumentNullException

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

GetTypeInfo(OpcEncoding)

Retrieves the OpcDataTypeInfo which declares the data type which is identified by the `encoding` specified.

C#

```
public static OpcDataTypeInfo GetTypeInfo(OpcEncoding encoding)
```

Parameters

`encoding` OpcEncoding

The OpcEncoding which identifies the OpcDataTypeInfo to retrieve.

Returns

OpcDataTypeInfo

The OpcDataTypeInfo 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).

GetTypeInfo(OpcNodeId)

Retrieves the OpcDataTypeInfo which declares the data type which is identified by the `typeId` specified.

C#

```
public static OpcDataTypeInfo GetTypeInfo(OpcNodeId typeId)
```

Parameters

`typeId` OpcNodeId

The OpcNodeId which identifies the OpcDataTypeInfo to retrieve.

Returns

OpcDataTypeInfo

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

Exceptions

ArgumentNullException

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

GetTypeInfo(Type)

Retrieves the OpcDataTypeInfo which declares the data type implemented by the `underlyingType` specified.

C#

```
public static OpcDataTypeInfo GetTypeInfo(Type underlyingType)
```

Parameters

`underlyingType` Type

The Type which implements the OpcDataTypeInfo to retrieve.

Returns

OpcDataTypeInfo

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

Exceptions

ArgumentNullException

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

GetTypeInfo<T>()

Retrieves the OpcDataTypeInfo which declares the data type implemented by the `T` specified.

C#

```
public static OpcDataTypeInfo GetTypeInfo<T>()
```

Returns

OpcDataTypeInfo

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

New(OpcDataTypeInfo)

Creates a new instance of the OpcDataObject class using the `dataType` specified.

C#

```
public static OpcDataObject New(OpcDataTypeInfo dataType)
```

Parameters

`dataType` OpcDataTypeInfo

The OpcDataTypeInfo which declares the type of data type instance the new OpcDataObject shall represent.

Returns

OpcDataObject

An instance of the OpcDataObject which can be used to dynamically access data type instance information.

Exceptions

ArgumentException

The `dataType` is not a structured data type (see IsStruct).

ArgumentNullException

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

NewArray(OpcDataTypeInfo, Int32)

Creates a one-dimensional array of OpcDataObject objects using the `dataType` and `length` with zero-based indexing.

C#

```
public static OpcDataObject[] NewArray(OpcDataTypeInfo dataType, int length)
```

Parameters

`dataType` OpcDataTypeInfo

The OpcDataTypeInfo which declares the type of data type instance the new OpcDataObject shall represent.

`length` Int32

An size of the array to create.

Returns

OpcDataObject[]

A new one-dimensional array of OpcDataObject instances which can be used to dynamically access array-based data type information

Exceptions

ArgumentException

The `dataType` is not a structured data type (see IsStruct).

ArgumentNullException

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

RegisterType(Type)

Registers the `dataType` specified as the data object type to create in case there a data instance is to be created for the type definition identified by the OpcDataTypeAttribute defined on the `dataType` specified.

C#

```
public static void RegisterType(Type dataType)
```

Parameters

`dataType` Type

The Type to use in case there a data object for the data type definition equals to the data type identifier defined using the OpcDataTypeAttribute is to be created. The type needs to provide either a public accessible parameterless constructor or a public accessible constructor accepting an instance of the IOpcDataStore interface and then needs to be a subclass of the OpcData class.

Exceptions

ArgumentException

The `dataType` does not define a public accessible parameterless constructor or a public accessible constructor accepting an instance of the IOpcDataStore interface or any data type specific attribute offers invalid information about the data type to associate with the Type.

ArgumentNullException

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

RegisterType(Type, OpcNodeId)

Registers the `dataType` specified as the data object type to create in case there a data instance is to be created for the type definition identified by the `dataTypeId` specified.

C#

```
public static void RegisterType(Type dataType, OpcNodeId dataTypeId)
```

Parameters

`dataType` Type

The Type to use in case there a data object for the data type definition equals to `dataTypeId` is to be created. The type needs to provide either a public accessible parameterless constructor or a public accessible constructor accepting an instance of the IOpcDataStore interface and then needs to be a subclass of the OpcData class.

`dataTypeId` OpcNodeId

The identifier of the type definition for which the `dataType` is to be used to create a data object instance.

Exceptions

ArgumentException

The `dataType` does not define a public accessible parameterless constructor or a public accessible constructor accepting an instance of the IOpcDataStore interface or any data type specific attribute offers invalid information about the data type to associate with the Type.

ArgumentNullException

The `dataType` or `dataTypeId` is a null reference (Nothing in Visual Basic).

Remarks

It is not possible to register the same or a different data type for the `dataTypeId` specified as long there is already a data type registered for the type definition specified by `dataTypeId`.

RegisterType(Type, OpcNodeId, OpcEncoding[])

Registers the `dataType` specified as the data object type to create in case there a data instance is to be created for the type definition identified by the `dataTypeId` or by one of the `dataTypeEncodings` specified.

C#

```
public static void RegisterType(Type dataType, OpcNodeId dataTypeId, params OpcEncoding[]
dataTypeEncodings)
```

Parameters

`dataType` Type

The Type to use in case there a data object for the data type definition equals to the data type identifier defined using the OpcDataTypeAttribute is to be created. The type needs to provide either a public accessible parameterless constructor or a public accessible constructor accepting an instance of the IOpcDataStore interface and then needs to be a subclass of the OpcData class.

`dataTypeId` OpcNodeId

The identifier of the type definition for which the `dataType` is to be used to create a data object instance.

`dataTypeEncodings` OpcEncoding[]

An array of OpcEncoding instances identifying the different encodings supported by the `dataType` specified. Using OpcEncoding instances enables the possibility to define OpcEncodingType specific identifiers to use to identify the according encoding information required to encode or decode the data object information.

Exceptions

ArgumentException

The type `dataType` does not define a public accessible parameterless constructor or a public accessible constructor accepting an instance of the IOpcDataStore interface or any data type specific attribute offers invalid information about the data type to associate with the Type.

ArgumentNullException

The `dataType`, `dataTypeId` or `dataTypeEncodings` or one of its items is a null reference (Nothing in Visual Basic).

RegisterType<T>()

Registers the type defined by `T` as the data object type to create in case there a data instance is to be created for the type definition identified by the OpcDataTypeAttribute defined on the `T` specified.

C#

```
public static void RegisterType<T>()
```

Exceptions

ArgumentException

The type `T` does not define a public accessible parameterless constructor or a public accessible constructor accepting an instance of the IOpcDataStore interface or any data type specific attribute offers invalid information about the data type to associate with the Type.

RegisterType<T>(OpcNodeId)

Registers the type defined by `T` as the data object type to create in case there a data instance is to be created for the type definition identified by the `dataTypeId` specified.

C#

```
public static void RegisterType<T>(OpcNodeId dataTypeId)
```

Parameters

`dataTypeId` OpcNodeId

The identifier of the type definition for which the `T` is to be used to create a data object instance.

Exceptions

ArgumentException

The type `T` does not define a public accessible parameterless constructor or a public accessible constructor accepting an instance of the IOpcDataStore interface or any data type specific attribute offers invalid information about the data type to associate with the Type.

ArgumentNullException

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

Remarks

It is not possible to register the same or a different data type for the `dataTypeId` specified as long there is already a data type registered for the type definition specified by `dataTypeId`.

RegisterType<T>(OpcNodeId, OpcEncoding[])

Registers the type defined by `T` as the data object type to create in case there a data instance is to be created for the type definition identified by the `dataTypeId` or by one of the `dataTypeEncodings` specified.

C#

```
public static void RegisterType<T>(OpcNodeId dataTypeId, params OpcEncoding[]
dataTypeEncodings)
```

Parameters

`dataTypeId` OpcNodeId

The identifier of the type definition for which the `T` is to be used to create a data object instance.

`dataTypeEncodings` OpcEncoding[]

An array of OpcEncoding instances identifying the different encodings supported by the `T` specified. Using OpcEncoding instances enables the possibility to define OpcEncodingType specific identifiers to use to identify the according encoding information required to encode or decode the data object information.

Exceptions

ArgumentException

The type `T` does not define a public accessible parameterless constructor or a public accessible constructor accepting an instance of the IOpcDataStore interface or any data type specific attribute offers invalid information about the data type to associate with the Type.

ArgumentNullException

The `dataTypeId` or `dataTypeEncodings` or one of its items is a null reference (Nothing in Visual Basic).

RegisterTypes(Assembly)

Registers all non-abstract types in the `assembly` specified which defines the OpcDataTypeAttribute.

C#

```
public static void RegisterTypes(Assembly assembly)
```

Parameters

`assembly` Assembly

The Assembly its types are inspected for types which using the OpcDataTypeAttribute.

Exceptions

ArgumentException

A Type does not define a public accessible parameterless constructor or a public accessible constructor accepting an instance of the IOpcDataStore interface or any data type specific attribute offers invalid information about the data type to associate with the Type.

ArgumentNullException

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

Remarks

The Type instances to register need to define the OpcDataTypeAttribute and either a public accessible parameterless constructor or a public accessible constructor accepting an instance of the IOpcDataStore interface and then needs to be a subclass of the OpcData class.

RegisterTypes(IEnumerable<Type>)

Registers the `dataTypes` specified as the data object type to create in case there a data instance is to be created for the type definitions identified by the OpcEventTypeAttribute used on one of the `dataTypes` specified.

C#

```
public static void RegisterTypes(IEnumerable<Type> dataTypes)
```

Parameters

`dataTypes` IEnumerable<Type>

A sequence of Type instances representing the types to use in case there a data object for the data

definition equals to the data type identifier defined using the OpcDataTypeAttribute is to be created.

Exceptions

ArgumentException

A Type does not define a public accessible parameterless constructor or a public accessible constructor accepting an instance of the IOpcDataStore interface or any data type specific attribute offers invalid information about the data type to associate with the Type.

ArgumentNullException

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

Remarks

Any null reference (Nothing in Visual Basic) in `dataTypes` is ignored.

RegisterTypes(Type, Type[])

Registers the `dataType` and the further specified `dataTypes` as the data object type to create in case there a data instance is to be created for the type definition identified by the OpcDataTypeAttribute used on the `dataType` or on one of the `dataTypes` specified.

C#

```
public static void RegisterTypes(Type dataType, params Type[] dataTypes)
```

Parameters

`dataType` Type

The Type instance to use in case there a data object for the data definition equals to the data type identifier defined using the OpcDataTypeAttribute is to be created.

`dataTypes` Type[]

An array of additional Type instances to use in case there a data object for the data definitions equals to the data type identifier defined using the OpcDataTypeAttribute is to be created.

Exceptions

ArgumentException

A Type does not define a public accessible parameterless constructor or a public accessible constructor accepting an instance of the IOpcDataStore interface or any data type specific attribute offers invalid information about the data type to associate with the Type.

ArgumentNullException

The `dataType` or `dataTypes` is a null reference (Nothing in Visual Basic).

ShallowCopy()

C#

```
protected virtual object ShallowCopy()
```

Returns

Object

UnregisterType<T>()

Unregisters the previously registered (see RegisterType´´1) type defined by `T` as the Object type to not longer create for the type definition identified by the OpcDataTypeAttribute defined on the `T` specified.

C#

```
public static void UnregisterType<T>()
```

UnregisterType<T>(OpcNodeId)

Unregisters the previously registered (see RegisterType´´1(OpcNodeId)) type defined by `T` as the Object type to not longer create for the type definition identified by the `dataTypeId` specified.

C#

```
public static void UnregisterType<T>(OpcNodeId dataTypeId)
```

Parameters

`dataTypeId` OpcNodeId

The identifier of the type definition for which the `T` is to be unregistered.

Exceptions

ArgumentNullException

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

UnregisterType<T>(OpcNodeId, OpcEncoding[])

Unregisters the previously registered (see RegisterType´´1(OpcNodeId, OpcEncoding)) type defined by `T` as the Object type to not longer create for the type definition identified by the `dataTypeId` or by one of the `dataTypeEncodings` specified.

C#

```
public static void UnregisterType<T>(OpcNodeId dataTypeId, params OpcEncoding[]
dataTypeEncodings)
```

Parameters

`dataTypeId` OpcNodeId

The identifier of the type definition for which the `T` is to be unregistered.

`dataTypeEncodings` OpcEncoding[]

An array of OpcEncoding instances identifying the different encodings supported by the `T` specified for which the `T` is to be unregistered as well.

Exceptions

ArgumentNullException

The `dataTypeId` or `dataTypeEncodings` or one of its items is a null reference (Nothing in Visual Basic).

UnregisterTypes(Assembly)

Unregisters all previously registered types (see RegisterTypes(Assembly)) in the `assembly` specified.

C#

```
public static void UnregisterTypes(Assembly assembly)
```

Parameters

`assembly` Assembly

The Assembly its types are inspected for types using the OpcDataTypeAttribute.

Exceptions

ArgumentNullException

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

UnregisterTypes(IEnumerable<Type>)

Unregisters the previously registered (see RegisterTypes(IEnumerable)) `dataTypes` specified.

C#

```
public static void UnregisterTypes(IEnumerable<Type> dataTypes)
```

Parameters

`dataTypes` IEnumerable<Type>

A sequence of Type instances to unregister for the data definitions equals to the data type identifiers defined using the OpcDataTypeAttribute.

Exceptions

ArgumentNullException

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

UnregisterTypes(Type, Type[])

Unregisters the previously registered `dataType` and the further specified `dataTypes` (see RegisterTypes(Type, Type)).

C#

```
public static void UnregisterTypes(Type dataType, params Type[] dataTypes)
```

Parameters

`dataType` Type

The Type instance to unregister for the data definitions equals to the data type identifiers defined using the OpcDataTypeAttribute.

`dataTypes` Type[]

Exceptions

ArgumentNullException

The `dataType` or `dataTypes` is a null reference (Nothing in Visual Basic).

Table of Contents

Constructors 1
OpcData(IOpcDataStore) 1
Events 1
TypeResolve 1
Properties 1
DataStore 1
Methods 2
DeepCopy() 2
GetType(OpcNodeId) 2
GetTypeInfo(OpcEncoding) 3
GetTypeInfo(OpcNodeId) 3
GetTypeInfo(Type) 4
GetTypeInfo<T>() 4
New(OpcDataTypeInfo) 5
NewArray(OpcDataTypeInfo, Int32) 5
RegisterType(Type) 6
RegisterType(Type, OpcNodeId) 6
RegisterType(Type, OpcNodeId, OpcEncoding[]) 7
RegisterType<T>() 8
RegisterType<T>(OpcNodeId) 8
RegisterType<T>(OpcNodeId, OpcEncoding[]) 9
RegisterTypes(Assembly) 10
RegisterTypes(IEnumerable<Type>) 10
RegisterTypes(Type, Type[]) 11
ShallowCopy() 12
UnregisterType<T>() 12
UnregisterType<T>(OpcNodeId) 12
UnregisterType<T>(OpcNodeId, OpcEncoding[]) 12
UnregisterTypes(Assembly) 13
UnregisterTypes(IEnumerable<Type>) 13
UnregisterTypes(Type, Type[]) 14