

OpcEncoding Members

Namespace: Opc.UaFx

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

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

Properties

DefaultBinary

Gets the foundation defined encoding used by default to encode information using the binary representation (known as 'OPC UA Binary') of the information to serialize or deserialize.

C#

```
public static OpcEncoding DefaultBinary { get; }
```

Property Value

[OpcEncoding](#)

An instance of the [OpcEncoding](#) class used to define the OPC UA Binary encoding. The [OpcEncoding](#) instance provided identifies a data independent encoding and does therefore not refer to a specific node which refers to the [OpcEncodingType](#) used by the instance ([Id](#) is equals [Null](#)).

DefaultXml

Gets the foundation defined encoding used by default to encode information using a XML-based serialization format (known as 'OPC UA XML') to serialize or deserialize information.

C#

```
public static OpcEncoding DefaultXml { get; }
```

Property Value

[OpcEncoding](#)

An instance of the [OpcEncoding](#) class used to define the OPC UA XML encoding. The [OpcEncoding](#) instance provided identifies a data independent encoding and does therefore not refer to a specific node which refers to the [OpcEncodingType](#) used by the instance ([Id](#) is equals [Null](#)).

Id

Gets the identifier of the node which refers to a specific [OpcEncodingType](#) (defined by [Type](#)) used to encode information.

C#

```
public OpcNodeId Id { get; }
```

Property Value

OpcNodeId

An instance of the [OpcNodeId](#) class which either identifies a data type specific encoding or [Null](#) in case there this [OpcEncoding](#) represents one of the default encodings defined by the foundation.

Namespace

Gets the namespace within the encoding has been defined. This can be the same namespace as referenced by data type dictionaries using their [NamespaceUri](#) property.

C#

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

Property Value

OpcNamespace

An instance of the [OpcNamespace](#) class which refers to the namespace within the encoding has been defined. These instance may be the same [OpcNamespace](#) as used by the [Id](#) (see [Namespace](#)) or a custom namespace which in case of data type encodings have to be the same namespace as referenced by data type dictionaries using their [NamespaceUri](#) property.

Type

Gets a value which identifies the underlying logic used to encode information.

C#

```
public OpcEncodingType Type { get; }
```

Property Value

OpcEncodingType

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

Methods

All(OpcNodeId)

Retrieves for each supported [OpcEncodingType](#) a new [OpcEncoding](#) while each uses the [id](#) specified.

C#

```
public static OpcEncoding[] All(OpcNodeId id)
```

Parameters

id [OpcNodeId](#)

The identifier of the node which relates to the encodings offered.

Returns

[OpcEncoding\[\]](#)

An array of [OpcEncoding](#) instances representing all [OpcEncodingTypes](#) supported while referencing to the encoding node specified by **id**.

Exceptions

[ArgumentNullException](#)

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

All(OpcNodeId, OpcNamespace)

Retrieves for each supported [OpcEncodingType](#) a new [OpcEncoding](#) while each uses the **id** and **encodingNamespace** specified.

C#

```
public static OpcEncoding[] All(OpcNodeId id, OpcNamespace encodingNamespace)
```

Parameters

id [OpcNodeId](#)

The identifier of the node which relates to the encodings offered.

encodingNamespace [OpcNamespace](#)

The [OpcNamespace](#) which refers to the namespace within the encodings has been defined. In case of data type encodings this [OpcNamespace](#) instance have to be the same namespace as referenced by data type dictionaries using their NamespaceUri property. This parameter can be a null reference (Nothing in Visual Basic); in this case the [Namespace](#) of the **id** specified is used instead.

Returns

[OpcEncoding\[\]](#)

An array of [OpcEncoding](#) instances representing all [OpcEncodingTypes](#) supported while referencing to the encoding node specified by **id**.

Exceptions

ArgumentNullException

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

Binary(OpcEncoding[])

Selects the first `OpcEncoding` in `encodings` which defines an encoding which uses the `OpcEncodingTypeBinary` as the logic used to encode data.

C#

```
public static OpcEncoding Binary(params OpcEncoding[] encodings)
```

Parameters

`encodings` `OpcEncoding[]`

An array of `OpcEncoding` instances from which the first binary encoding is to be selected.

Returns

`OpcEncoding`

The first binary `OpcEncoding` in `encodings` or a null reference (Nothing in Visual Basic)

Binary(OpcNodeId)

Retrieves a new `OpcEncoding` which uses the `id` specified to define an encoding which uses the logic used for binary data encoding.

C#

```
public static OpcEncoding Binary(OpcNodeId id)
```

Parameters

`id` `OpcNodeId`

The identifier of the node which relates to the encoding offered.

Returns

`OpcEncoding`

A new instance of the `OpcEncoding` class representing the encoding node specified by the `id` while using the `BinaryOpcEncodingType` as the `Type` of encoding.

Exceptions

ArgumentNullException

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

Binary(OpcNodeId, OpcNamespace)

Retrieves a new `OpcEncoding` which uses the `id` and `encodingNamespace` specified to define an encoding which uses the logic used for binary data encoding.

C#

```
public static OpcEncoding Binary(OpcNodeId id, OpcNamespace encodingNamespace)
```

Parameters

`id` `OpcNodeId`

The identifier of the node which relates to the encoding offered.

`encodingNamespace` `OpcNamespace`

The `OpcNamespace` which refers to the namespace within the encoding has been defined. In case of data type encodings this `OpcNamespace` instance have to be the same namespace as referenced by data type dictionaries using their `NamespaceUri` property. This parameter can be a null reference (Nothing in Visual Basic); in this case the `Namespace` of the `id` specified is used instead.

Returns

`OpcEncoding`

A new instance of the `OpcEncoding` class representing the encoding node specified by the `id` within the further defined `encodingNamespace` while using the `BinaryOpcEncodingType` as the `Type` of encoding.

Exceptions

`ArgumentNullException`

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

Default(OpcEncodingType)

Retrieves the according default `OpcEncoding` which is used for the `encodingType` specified.

C#

```
public static OpcEncoding Default(OpcEncodingType encodingType)
```

Parameters

`encodingType` `OpcEncodingType`

The `OpcEncodingType` for that the according default `OpcEncoding` is to be determined.

Returns

OpcEncoding

The default [OpcEncoding](#) used for the `encodingType` specified.

Equals(Object)

Determines whether the specified `other` is equal to this [OpcEncoding](#).

C#

```
public override bool Equals(object other)
```

Parameters

`other` [Object](#)

The [OpcEncoding](#) to compare to the current [OpcEncoding](#).

Returns

Boolean

The value true if the specified [OpcEncoding](#) is equal to the current [OpcEncoding](#); otherwise the value false.

Equals(OpcEncoding)

Determines whether the specified `other` is equal to this [OpcEncoding](#).

C#

```
public bool Equals(OpcEncoding other)
```

Parameters

`other` [OpcEncoding](#)

The [OpcEncoding](#) to compare to the current [OpcEncoding](#).

Returns

Boolean

The value true if the specified [OpcEncoding](#) is equal to the current [OpcEncoding](#); otherwise the value false.

GetHashCode()

Retrieves a hash code for this [OpcEncoding](#).

C#

```
public override int GetHashCode()
```

Returns

Int32

An Int32 that contains the hash code for the OpcEncoding.

GetType(String)

Determines the expressed type of encoding using the information encoded in the encodingName specified.

C#

```
public static OpcEncodingType? GetType(string encodingName)
```

Parameters

encodingName String

A String expressing the name of a specific encoding from which the according OpcEncodingType is to be determined.

Returns

Nullable<OpcEncodingType>

One of the members defined by the OpcEncodingType enumeration or a null reference (Nothing in Visual Basic) in case there no encoding type related information could be determined in the encodingName.

Of(OpcEncodingType, OpcEncoding[])

Selects the first OpcEncoding in encodings which defines an encoding which uses the encodingType specified as the logic used to encode data.

C#

```
public static OpcEncoding[] Of(OpcEncodingType encodingType, params OpcEncoding[] encodings)
```

Parameters

encodingType OpcEncodingType

The type of encoding the OpcEncoding instance is to be selected from the encodings specified.

encodings OpcEncoding[]

An array of OpcEncoding instances from which the first encoding with of the encodingType specified is to be selected.

Returns

OpcEncoding[]

The first [OpcEncoding](#) in [encodings](#) of the [encodingType](#) specified or a null reference (Nothing in Visual Basic).

ToString()

Returns a string representing the encoding described.

C#

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

Returns

[String](#)

A string represening the information used by this [OpcEncoding](#) instance.

Xml(OpcEncoding[])

Selects the first [OpcEncoding](#) in [encodings](#) which defines an encoding which uses the [OpcEncodingTypeXml](#) as the logic used to encode data.

C#

```
public static OpcEncoding Xml(params OpcEncoding[] encodings)
```

Parameters

[encodings](#) [OpcEncoding\[\]](#)

An array of [OpcEncoding](#) instances from which the first XML encoding is to be selected.

Returns

[OpcEncoding](#)

The first XML [OpcEncoding](#) in [encodings](#) or a null reference (Nothing in Visual Basic)

Xml(OpcNodeId)

Retrieves a new [OpcEncoding](#) which uses the [id](#) specified to define an encoding which uses the logic used for XML data encoding.

C#

```
public static OpcEncoding Xml(OpcNodeId id)
```

Parameters

id OpcNodeId

The identifier of the node which relates to the encoding offered.

Returns

OpcEncoding

A new instance of the [OpcEncoding](#) class representing the encoding node specified by the [id](#) while using the [XmlOpcEncodingType](#) as the [Type](#) of encoding.

Exceptions

ArgumentNullException

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

Xml(OpcNodeId, OpcNamespace)

Retrieves a new [OpcEncoding](#) which uses the [id](#) and [encodingNamespace](#) specified to define an encoding which uses the logic used for XML data encoding.

C#

```
public static OpcEncoding Xml(OpcNodeId id, OpcNamespace encodingNamespace)
```

Parameters

id OpcNodeId

The identifier of the node which relates to the encoding offered.

encodingNamespace OpcNamespace

The [OpcNamespace](#) which refers to the namespace within the encoding has been defined. In case of data type encodings this [OpcNamespace](#) instance have to be the same namespace as referenced by data type dictionaries using their NamespaceUri property. This parameter can be a null reference (Nothing in Visual Basic); in this case the [Namespace](#) of the [id](#) specified is used instead.

Returns

OpcEncoding

A new instance of the [OpcEncoding](#) class representing the encoding node specified by the [id](#) within the further defined [encodingNamespace](#) while using the [XmlOpcEncodingType](#) as the [Type](#) of encoding.

Exceptions

ArgumentNullException

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

Operators

Equality(OpcEncoding, OpcEncoding)

Returns a value indicating whether two instance of [OpcEncoding](#) are equal.

C#

```
public static bool operator ==(OpcEncoding left, OpcEncoding right)
```

Inequality(OpcEncoding, OpcEncoding)

Returns a value indicating whether two instances of [OpcEncoding](#) are not equal.

C#

```
public static bool operator !=(OpcEncoding left, OpcEncoding right)
```

Table of Contents

Properties	1
DefaultBinary	1
DefaultXml	1
Id	1
Namespace	2
Type	2
Methods	2
All(OpcNodeId)	2
All(OpcNodeId, OpcNamespace)	3
Binary(OpcEncoding[])	4
Binary(OpcNodeId)	4
Binary(OpcNodeId, OpcNamespace)	5
Default(OpcEncodingType)	5
Equals(Object)	6
Equals(OpcEncoding)	6
GetHashCode()	6
GetType(String)	7
Of(OpcEncodingType, OpcEncoding[])	7
ToString()	8
Xml(OpcEncoding[])	8
Xml(OpcNodeId)	8
Xml(OpcNodeId, OpcNamespace)	9
Operators	10
Equality(OpcEncoding, OpcEncoding)	10
Inequality(OpcEncoding, OpcEncoding)	10

