

OpcNodeId Members

Namespace: Opc.UaFx

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

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

Constructors

OpcNodeId(Byte[])

Initializes a new instance of the [OpcNodeId](#) class using the **value** specified.

C#

```
public OpcNodeId(byte[] value)
```

Parameters

value Byte[]

The opaque value of the identifier to represent. A value equals a null reference (Nothing in Visual Basic) or an array with the length equals to zero results into a null identifier (see [IsNull](#)).

OpcNodeId(Byte[], Int32)

Initializes a new instance of the [OpcNodeId](#) class using the **value** and **namespaceIndex** specified.

C#

```
public OpcNodeId(byte[] value, int namespaceIndex)
```

Parameters

value Byte[]

The opaque value of the identifier to represent. A value equals a null reference (Nothing in Visual Basic) or an array with the length equals to zero results into a null identifier (see [IsNull](#)).

namespaceIndex Int32

The index of the namespace that this identifier should refer to.

OpcNodeId(Byte[], OpcNamespace)

Initializes a new instance of the [OpcNodeId](#) class using the **value** and **nodeNamespace** specified.

C#

```
public OpcNodeId(byte[] value, OpcNamespace nodeNamespace)
```

Parameters

value Byte[]

The opaque value of the identifier to represent. A value equals a null reference (Nothing in Visual Basic) or an array with the length equals to zero results into a null identifier (see [IsNull](#)).

nodeNamespace OpcNamespace

The namespace that this identifier should refer to.

OpcNodeId(Guid)

Initializes a new instance of the [OpcNodeId](#) class using the **value** specified.

C#

```
public OpcNodeId(Guid value)
```

Parameters

value Guid

The general unique identifier (= GUID) to use as the value of the identifier to represent. A value equals [Empty](#) results into a null identifier (see [IsNull](#)).

OpcNodeId(Guid, Int32)

Initializes a new instance of the [OpcNodeId](#) class using the **value** and **namespaceIndex** specified.

C#

```
public OpcNodeId(Guid value, int namespaceIndex)
```

Parameters

value Guid

The general unique identifier (= GUID) to use as the value of the identifier to represent. A value equals [Empty](#) results into a null identifier (see [IsNull](#)).

namespaceIndex Int32

The index of the namespace that this identifier should refer to.

OpcNodeId(Guid, OpcNamespace)

Initializes a new instance of the [OpcNodeId](#) class using the **value** and **nodeNamespace** specified.

C#

```
public OpcNodeId(Guid value, OpcNamespace nodeNamespace)
```

Parameters

value Guid

The general unique identifier (= GUID) to use as the value of the identifier to represent. A value equals **Empty** results into a null identifier (see [IsNull](#)).

nodeNamespace [OpcNamespace](#)

The namespace that this identifier should refer to.

OpcNodeId(Int32)

Initializes a new instance of the [OpcNodeId](#) class using the **value** specified.

C#

```
public OpcNodeId(int value)
```

Parameters

value Int32

The numeric value of the identifier to represent. A value equals zero results into a null identifier (see [IsNull](#)).

OpcNodeId(Int32, Int32)

Initializes a new instance of the [OpcNodeId](#) class using the **value** and **namespaceIndex** specified.

C#

```
public OpcNodeId(int value, int namespaceIndex)
```

Parameters

value Int32

The numeric value of the identifier to represent. A value equals zero results into a null identifier (see [IsNull](#)).

namespaceIndex Int32

The index of the namespace that this identifier should refer to.

OpcNodeId(Int32, OpcNamespace)

Initializes a new instance of the [OpcNodeId](#) class using the **value** and **nodeNamespace** specified.

C#

```
public OpcNodeId(int value, OpcNamespace nodeNamespace)
```

Parameters

value [Int32](#)

The numeric value of the identifier to represent. A value equals zero results into a null identifier (see [IsNull](#)).

nodeNamespace [OpcNamespace](#)

The namespace that this identifier should refer to.

OpcNodeId(OpcNodeId)

Initializes a new instance of the [OpcNodeId](#) class using the [other](#) specified.

C#

```
public OpcNodeId(OpcNodeId other)
```

Parameters

other [OpcNodeId](#)

The [OpcNodeId](#) from that a deep copy is to be created.

Exceptions

[ArgumentNullException](#)

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

OpcNodeId(String)

Initializes a new instance of the [OpcNodeId](#) class using the [value](#) specified.

C#

```
public OpcNodeId(string value)
```

Parameters

value [String](#)

The [String](#) value of the identifier to represent. A null reference (Nothing in Visual Basic) or a value equals [Empty](#) results into a null identifier (see [IsNull](#)).

OpcNodeId(String, Int32)

Initializes a new instance of the [OpcNodeId](#) class using the [value](#) and [namespaceIndex](#) specified.

C#

```
public OpcNodeId(string value, int namespaceIndex)
```

Parameters

value String

The String value of the identifier to represent. A null reference (Nothing in Visual Basic) or a value equals Empty results into a null identifier (see IsNull).

namespaceIndex Int32

The index of the namespace that this identifier should refer to.

OpcNodeId(String, Int32, OpcName[])

Initializes a new instance of the OpcNodeId class using the value, namespaceIndex and pathElements specified.

C#

```
public OpcNodeId(string value, int namespaceIndex, params OpcName[] pathElements)
```

Parameters

value String

The String value of the identifier to represent. A null reference (Nothing in Visual Basic) or a value equals Empty results into a null identifier (see IsNull).

namespaceIndex Int32

The index of the namespace that this identifier should refer to.

pathElements OpcName[]

The elements of the OpcNamePath to use as the logical (= physically not available in the address space) portion of the identifier.

OpcNodeId(String, Int32, OpcNamePath)

Initializes a new instance of the OpcNodeId class using the value, namespaceIndex and path specified.

C#

```
public OpcNodeId(string value, int namespaceIndex, OpcNamePath path)
```

Parameters

value String

The String value of the identifier to represent. A null reference (Nothing in Visual Basic) or a value equals Empty results into a null identifier (see IsNull).

namespaceIndex Int32

The index of the namespace that this identifier should refer to.

path [OpcNamePath](#)

The logical (= physically not available in the address space) portion of the identifier.

OpcNodeId(String, OpcName[])

Initializes a new instance of the [OpcNodeId](#) class using the [value](#) and [pathElements](#) specified.

C#

```
public OpcNodeId(string value, params OpcName[] pathElements)
```

Parameters

[value](#) [String](#)

The [String](#) value of the identifier to represent. A null reference (Nothing in Visual Basic) or a value equals [Empty](#) results into a null identifier (see [IsNull](#)).

[pathElements](#) [OpcName\[\]](#)

The elements of the [OpcNamePath](#) to use as the logical (= physically not available in the address space) portion of the identifier.

OpcNodeId(String, OpcNamePath)

Initializes a new instance of the [OpcNodeId](#) class using the [value](#) and [path](#) specified.

C#

```
public OpcNodeId(string value, OpcNamePath path)
```

Parameters

[value](#) [String](#)

The [String](#) value of the identifier to represent. A null reference (Nothing in Visual Basic) or a value equals [Empty](#) results into a null identifier (see [IsNull](#)).

[path](#) [OpcNamePath](#)

The logical (= physically not available in the address space) portion of the identifier.

OpcNodeId(String, OpcNamespace)

Initializes a new instance of the [OpcNodeId](#) class using the [value](#) and [nodeNamespace](#) specified.

C#

```
public OpcNodeId(string value, OpcNamespace nodeNamespace)
```

Parameters

value String

The String value of the identifier to represent. A null reference (Nothing in Visual Basic) or a value equals Empty results into a null identifier (see IsNull).

nodeNamespace OpcNamespace

The namespace that this identifier should refer to.

OpcNodeId(String, OpcNamespace, OpcName[])

Initializes a new instance of the OpcNodeId class using the value, nodeNamespace and pathElements specified.

C#

```
public OpcNodeId(string value, OpcNamespace nodeNamespace, params OpcName[] pathElements)
```

Parameters

value String

The String value of the identifier to represent. A null reference (Nothing in Visual Basic) or a value equals Empty results into a null identifier (see IsNull).

nodeNamespace OpcNamespace

The namespace that this identifier should refer to.

pathElements OpcName[]

The elements of the OpcNamePath to use as the logical (= physically not available in the address space) portion of the identifier.

OpcNodeId(String, OpcNamespace, OpcNamePath)

Initializes a new instance of the OpcNodeId class using the value, nodeNamespace and path specified.

C#

```
public OpcNodeId(string value, OpcNamespace nodeNamespace, OpcNamePath path)
```

Parameters

value String

The String value of the identifier to represent. A null reference (Nothing in Visual Basic) or a value equals Empty results into a null identifier (see IsNull).

nodeNamespace OpcNamespace

The namespace that this identifier should refer to.

path OpcNamePath

The logical (= physically not available in the address space) portion of the identifier.

OpcNodeId(UInt32)

Initializes a new instance of the [OpcNodeId](#) class using the **value** specified.

C#

```
[CLSCompliant(false)]  
public OpcNodeId(uint value)
```

Parameters

value [UInt32](#)

The numeric value of the identifier to represent. A value equals zero results into a null identifier (see [IsNull](#)).

OpcNodeId(UInt32, Int32)

Initializes a new instance of the [OpcNodeId](#) class using the **value** and **namespaceIndex** specified.

C#

```
[CLSCompliant(false)]  
public OpcNodeId(uint value, int namespaceIndex)
```

Parameters

value [UInt32](#)

The numeric value of the identifier to represent. A value equals zero results into a null identifier (see [IsNull](#)).

namespaceIndex [Int32](#)

The index of the namespace that this identifier should refer to.

OpcNodeId(UInt32, OpcNamespace)

Initializes a new instance of the [OpcNodeId](#) class using the **value** and **nodeNamespace** specified.

C#

```
[CLSCompliant(false)]  
public OpcNodeId(uint value, OpcNamespace nodeNamespace)
```

Parameters

value [UInt32](#)

The numeric value of the identifier to represent. A value equals zero results into a null identifier (see

IsNull).

nodeNamespace [OpcNamespace](#)

The namespace that this identifier should refer to.

Fields

DefaultSeparator

Defines the character which is used to separate parent-child related node identifiers from each other when constructing a parent-child related node identifier using [Of\(OpcName, OpcNodeid\)](#).

C#

```
public static readonly char DefaultSeparator
```

Field Value

[Char](#)

DefaultSeparatorOfPath

Defines the character which is used to separate node-component-node related node identifiers from each other when constructing a node-component-node related node identifier using [WithPath\(OpcNamePath\)](#).

C#

```
public static readonly char DefaultSeparatorOfPath
```

Field Value

[Char](#)

Properties

Factory

Gets or sets the globally used [OpcNodeidFactory](#) to generate new instances of the [OpcNodeid](#) class.

C#

```
public static OpcNodeidFactory Factory { get; set; }
```

Property Value

[OpcNodeidFactory](#)

An instance of a sub class of the [OpcNodeidFactory](#) class. The default factory used is the

OpcNominalNodeIdFactory.

IsAbsolute

Gets a value indicating whether the identifier is an absolute one.

C#

```
public bool IsAbsolute { get; }
```

Property Value

Boolean

The value true if the [Namespace](#) is an absolute namespace and is therefore not server dependent upon its [Index](#); otherwise the value false.

IsNull

Gets a value indicating whether the node identifier is a null identifier.

C#

```
public bool IsNull { get; }
```

Property Value

Boolean

The value true if the [Value](#) defines a [Type](#) specific definition of a null identifier; otherwise the value false.

Remarks

A null identifier originates when a node identifier is created using: * A [Byte](#)-array as the identifier value. In case there the [Opaque](#) identifier value is a null reference (Nothing in Visual Basic) or an empty array (the length is equals zero).

- A [Guid](#) as the identifier value. In case there the [Guid](#) identifier value is equals to [Empty](#).
- A [Int32](#) or [UInt32](#) as the identifier value. In case there the [Numeric](#) identifier value is equals to zero.
- A [String](#) as the identifier value. In case there the [String](#) identifier value is a null reference (Nothing in Visual Basic) or equals to [Empty](#).

Namespace

Gets the [OpcNamespace](#) used by the [OpcNodeId](#) to describe the namespace to that the node identifier belongs.

C#

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

Property Value

OpcNamespace

An instance of the [OpcNamespace](#) with the known information about the namespace to that the [OpcNodeId](#) belongs.

NamespaceIndex

Gets the index of the namespace that this identifier belongs.

C#

```
public int NamespaceIndex { get; }
```

Property Value

Int32

The index value which refers to the namespace in the namespace array of the server to that the node identifier belong.

Remarks

The available namespaces of the server can be retrieved through reading the value of the namespaces-node using the node identifier 'OpcObjectTypes.Server.Namespaces'.

NamespaceUri

Gets the uniform resource identifier (URI) of the namespace referred by the [NamespaceIndex](#).

C#

```
public Uri NamespaceUri { get; }
```

Property Value

Uri

The [Uri](#) to that the [OpcNodeId](#) refers using the [NamespaceIndex](#).

Null

Gets the default null node identifier.

C#

```
public static OpcNodeId Null { get; }
```

Property Value

OpcNodeId

An instance of the [OpcNodeId](#) which can be used for general purpose in cases there a null node identifier is enough.

OriginalFormat

Gets the original format of the node identifier from that the instance was constructed.

C#

```
public OpcNodeIdFormat OriginalFormat { get; }
```

Property Value

OpcNodeIdFormat

The format used by the [String](#) passed to the [Parse\(String\)](#) method from that the identifier has been constructed. If this instance has not been constructed through a [Parse\(String\)](#) call a constructor dependent [OpcNodeIdFormat](#) enumeration member.

OriginalString

Gets the original node identifier string that was passed to the [Parse\(String\)](#) method.

C#

```
public string OriginalString { get; }
```

Property Value

String

A [String](#) containing the exact node identifier specified from this instance was constructed; otherwise [Empty](#).

Path

Gets the path portion of the [Value](#) which refers to the logical (= physically not available in the address space) portion of the identifier.

C#

```
public OpcNamePath Path { get; }
```

Property Value

OpcNamePath

An instance of the [OpcNamePath](#) identifying the browse path to use to determine the according entity through the physically existing members in the address space.

Remarks

A [OpcNodeId](#) with a [OpcNamePath](#) not equals [Empty](#) identifies an entity which is not necessarily physically available in the address space. We define that the identifier of the physical ancestor of the entity identified by such an identifier (its logical owner which is physically available in the address space) is encoded in the [Value](#) of its [OpcNodeId](#) followed by the string formatted [Path](#). Excluding the [Path](#) from this [OpcNodeId](#) (see [ExcludePath](#)) results into the identifier of the 'physical' ancestor. Using these (path-less) identifier the ancestor can be looked up in the address space and can be used to browse for the entity using the [Path](#) of the identifier of its successor.

Type

Gets the type of [Value](#) used to define the identifier.

C#

```
public OpcNodeIdType Type { get; }
```

Property Value

[OpcNodeIdType](#)

One of the members defined by the [OpcNodeIdType](#) enumeration. The value depends on the type of [Value](#) used for the identifier.

Value

Gets the [Type](#) dependent value of the identifier represented.

C#

```
public object Value { get; }
```

Property Value

[Object](#)

A [Byte](#)-array, [Guid](#), [UInt32](#) or [String](#) value used to define the identifier.

ValueAsString

Gets the [Value](#) formatted as a [String](#).

C#

```
public string ValueAsString { get; }
```

Property Value

[String](#)

A [String](#) representing the textual representation of the [Value](#).

Methods

CompareTo(Object)

Compares the current [OpcNodeId](#) with the [other](#).

C#

```
public int CompareTo(object other)
```

Parameters

[other](#) [Object](#)

The [OpcNodeId](#) to compare with this [OpcNodeId](#).

Returns

[Int32](#)

A 32-bit signed integer that indicates the relative order of the objects being compared ([CompareTo\(Object\)](#)).

CompareTo(OpcNodeId)

Compares the current [OpcNodeId](#) with another [OpcNodeId](#).

C#

```
public int CompareTo(OpcNodeId other)
```

Parameters

[other](#) [OpcNodeId](#)

The [OpcNodeId](#) to compare with this [OpcNodeId](#).

Returns

[Int32](#)

A 32-bit signed integer that indicates the relative order of the objects being compared ([CompareTo\(\)](#)).

Equals(Object)

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

C#

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

Parameters

other Object

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

Returns

Boolean

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

Equals(OpcNodeId)

Determines whether the specified **other** is equal to this [OpcNodeId](#).

C#

```
public bool Equals(OpcNodeId other)
```

Parameters

other [OpcNodeId](#)

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

Returns

Boolean

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

Equals(OpcNodeId, OpcNodeId)

Determines whether two specified [OpcNodeId](#) objects are equals.

C#

```
public static bool Equals(OpcNodeId first, OpcNodeId second)
```

Parameters

first [OpcNodeId](#)

The first [OpcNodeId](#) to compare, or a null reference (Nothing in Visual Basic).

second [OpcNodeId](#)

The second [OpcNodeId](#) to compare, or a null reference (Nothing in Visual Basic).

Returns

Boolean

The value true if the **first** is equals to the **second**; otherwise the value false.

Equals(OpcNodeId, OpcNodeId, OpcNodeIdComparison)

Determines whether two specified **OpcNodeId** objects are equals according to a specific comparison method.

C#

```
public static bool Equals(OpcNodeId first, OpcNodeId second, OpcNodeIdComparison comparisonType)
```

Parameters

first OpcNodeId

The first **OpcNodeId** to compare, or a null reference (Nothing in Visual Basic).

second OpcNodeId

The second **OpcNodeId** to compare, or a null reference (Nothing in Visual Basic).

comparisonType OpcNodeIdComparison

One of the members defined by the **OpcNodeIdComparison** enumeration that specifies the rules for the comparison.

Returns

Boolean

The value true if the **first** is equals to the **second**; otherwise the value false.

Equals(OpcNodeId, OpcNodeIdComparison)

Determines whether the specified **OpcNodeId** object is equals to this instance according to a specific comparison method.

C#

```
public bool Equals(OpcNodeId other, OpcNodeIdComparison comparisonType)
```

Parameters

other OpcNodeId

The **OpcNodeId** to compare to this instance.

comparisonType OpcNodeIdComparison

One of the members defined by the **OpcNodeIdComparison** enumeration that specifies the rules for the

comparison.

Returns

Boolean

The value true if the **other** is equals to this instance; otherwise the value false.

ExcludePath()

Retrieves a new **OpcNodeId** there its **Value** not longer contains any **Path** information.

C#

```
public OpcNodeId ExcludePath()
```

Returns

OpcNodeId

A new instance of the **OpcNodeId** class using the same **Namespace** as this **OpcNodeId** and the same **Value**, but without any **Path** portion used by this **OpcNodeId**.

Remarks

In case there the identifier of the ancestor from that this **OpcNodeId** (with path) was created from is not of the **TypeString** this method will not recover its original **OpcNodeIdType**.

GetHashCode()

Retrieves a hash code for this **OpcNodeId**.

C#

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

Returns

Int32

An **Int32** that contains the hash code for the **OpcNodeId**.

IsNullOrEmpty(OpcNodeId)

Indicates whether the specified **OpcNodeId** is null or its **Value** is null, default or an empty string ("").

C#

```
public static bool IsNullOrEmpty(OpcNodeId identifier)
```

Parameters

identifier [OpcNodeId](#)

The [OpcNodeId](#) to test.

Returns

[Boolean](#)

The value true if the **identifier** parameter or its [Value](#) is null, default or an empty string (""); otherwise the value false.

Remarks

This method ignores the circumstance whether the [Namespace](#) of the **identifier** might be specified anyway.

Of(OpcName)

Creates a new [OpcNodeId](#) instance from the **name** specified.

C#

```
public static OpcNodeId Of(OpcName name)
```

Parameters

name [OpcName](#)

The [OpcName](#) its [Value](#) and [NamespaceIndex](#) is used to create a new [OpcNodeId](#) of the type [String](#).

Returns

[OpcNodeId](#)

A new instance of the [OpcNodeId](#) created from the [Value](#) and [NamespaceIndex](#) of the **name** specified.

Exceptions

[ArgumentException](#)

The **name** does not provide a [Value](#) nor a [NamespaceIndex](#) from that a [OpcNodeId](#) can be created (see [IsNull](#)).

[ArgumentNullException](#)

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

Of(OpcName, OpcNodeId)

Creates a new parent-child related [OpcNodeId](#) instance from the [name](#) and [parentNodeId](#) specified.

C#

```
public static OpcNodeId Of(OpcName name, OpcNodeId parentNodeId)
```

Parameters

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

The [OpcName](#) its [Value](#) and [NamespaceIndex](#) is used to create a new [OpcNodeId](#) of the type [String](#).

[parentNodeId](#) [OpcNodeId](#)

The [OpcNodeId](#) of the parent node its identifier is to be integrated into the new node identifier.

Returns

[OpcNodeId](#)

A new instance of the [OpcNodeId](#) created from the [Value](#) and [NamespaceIndex](#) of the [name](#) specified.

Exceptions

[ArgumentException](#)

The [name](#) does not provide a [Value](#) nor a [NamespaceIndex](#) from that a [OpcNodeId](#) can be created (see [IsNull](#)).

[ArgumentNullException](#)

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

Of(OpcName, OpcNodeId, Char)

Creates a new parent-child related [OpcNodeId](#) instance from the [name](#) and [parentNodeId](#) specified.

C#

```
public static OpcNodeId Of(OpcName name, OpcNodeId parentNodeId, char separator)
```

Parameters

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

The [OpcName](#) its [Value](#) and [NamespaceIndex](#) is used to create a new [OpcNodeId](#) of the type [String](#).

[parentNodeId](#) [OpcNodeId](#)

The [OpcNodeId](#) of the parent node its identifier is to be integrated into the new node identifier.

[separator](#) [Char](#)

The `Char` used to separate the child node identifier from the `parentNodeId`.

Returns

`OpcNodeId`

A new instance of the `OpcNodeId` created from the `Value` and `NamespaceIndex` of the `name` specified.

Exceptions

`ArgumentException`

The `name` does not provide a `Value` nor a `NamespaceIndex` from that a `OpcNodeId` can be created (see `IsNull`).

`ArgumentNullException`

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

Remarks

In case there the `parentNodeId` ends with the `separator` specified it will be not adopted by the newly created node identifier.

Parse(String)

Converts a node identifier string to a `OpcNodeId` instance.

C#

```
public static OpcNodeId Parse(string value)
```

Parameters

`value` `String`

A string that contains a node identifier.

Returns

`OpcNodeId`

An instance of the `OpcNodeId` class.

Exceptions

`FormatException`

The `value` is not a valid node identifier.

Resolve(IOpcNamespaceResolver)

Resolves this node identifier to a new node identifier its [Namespace](#) might be known in the context of the [resolver](#) object specified.

C#

```
public OpcNodeId Resolve(IOpcNamespaceResolver resolver)
```

Parameters

[resolver](#) [IOpcNamespaceResolver](#)

An instance implementing the [IOpcNamespaceResolver](#) interface to use to retrieve the namespace known using the namespace information of this identifier.

Returns

[OpcNodeId](#)

A deep copy of this [OpcNodeId](#) using either the resolved namespace or the same namespace as this node identifier.

Exceptions

[ArgumentNullException](#)

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

ToString()

Returns a string representing the node identifier.

C#

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

Returns

[String](#)

A string formatted using the [OriginalFormat](#) from that the [OpcNodeId](#) has been constructed. The format depends on the format detected through calling [Parse\(String\)](#) or the format defined using a specific constructor.

ToString(OpcNodeIdFormat)

Returns a string representing the node identifier in the [format](#) specified.

C#

```
public string ToString(OpcNodeIdFormat format)
```

Parameters

format [OpcNodeIdFormat](#)

One of the formatting styles defined by the [OpcNodeIdFormat](#) enumeration.

Returns

[String](#)

A string formatted using the **format** specified.

TryParse(String, out OpcNodeId)

Determines whether a string is a valid node identifier.

C#

```
public static bool TryParse(string value, out OpcNodeId nodeId)
```

Parameters

value [String](#)

The string to validate.

nodeId [OpcNodeId](#)

The [OpcNodeId](#) version of the string.

Returns

[Boolean](#)

The value true, if **value** is a valid node identifier; otherwise the value false.

UriIdentifier(OpcNodeId)

Retrieves the [Value](#) of the **nodeId** as an [Uri](#).

C#

```
public static Uri UriIdentifier(OpcNodeId nodeId)
```

Parameters

nodeId [OpcNodeId](#)

The [OpcNodeId](#) from its [Value](#) an [Uri](#) is to be retrieved.

Returns

Uri

An [Uri](#) if the [Type](#) of the [nodeId](#) is equals [String](#) and the [Value](#) can be parsed to an [Uri](#); otherwise a null reference (Nothing in Visual Basic).

Exceptions

ArgumentNullException

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

Remarks

This method can be used in scenarios there a server uses [Uri](#)-formatted strings as the identifier for its nodes, while its node managers defines different namespaces where the namespaceUris define base-uri's of the node identifier uri's. This allows the framework to determine the manager of a node without the need to know the concrete index of the namespace which defines the identifier of a node.

WithPath(OpcName[])

Defines a new [OpcNodeId](#) using the [Value](#), [Namespace](#) and [Path](#) of this [OpcNodeId](#) by combining this [Path](#) with the [pathElements](#) specified.

C#

```
public OpcNodeId WithPath(params OpcName[] pathElements)
```

Parameters

[pathElements](#) [OpcName\[\]](#)

The elements of the [OpcNamePath](#) to use as the logical (= physically not available in the address space) portion to append to the [Path](#) of this [OpcNodeId](#) for the identifier to create.

Returns

OpcNodeId

A new [OpcNodeId](#) derived from this [OpcNodeId](#) by extending the [Path](#) using the [pathElements](#) specified.

Remarks

In case there this [OpcNodeId](#) is not of the [TypeString](#) the new one will be, because of the [Path](#) is encoded in the [Value](#).

WithPath(OpcNamePath)

Defines a new [OpcNodeId](#) using the [Value](#), [Namespace](#) and [Path](#) of this [OpcNodeId](#) by combining this [Path](#)

with the **path** specified.

C#

```
public OpcNodeId WithPath(OpcNamePath path)
```

Parameters

path [OpcNamePath](#)

The logical (= physically not available in the address space) portion to append to the [Path](#) of this [OpcNodeId](#) for the identifier to create.

Returns

[OpcNodeId](#)

A new [OpcNodeId](#) derived from this [OpcNodeId](#) by extending the [Path](#) using the **path** specified.

Remarks

In case there this [OpcNodeId](#) is not of the [TypeString](#) the new one will be, because of the [Path](#) is encoded in the [Value](#).

Operators

Equality(OpcNodeId, OpcNodeId)

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

C#

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

Explicit(OpcNodeId to ExpandedNodeId)

Converts a [OpcNodeId](#) to an [ExpandedNodeId](#) object.

C#

```
[CLSCompliant(false)]  
public static explicit operator ExpandedNodeId(OpcNodeId value)
```

Explicit(OpcNodeId to NodeId)

Converts a [OpcNodeId](#) to an [NodeId](#) object.

C#


```
[CLSCompliant(false)]  
public static explicit operator NodeId(OpcNodeId value)
```

GreaterThan(OpcNodeId, OpcNodeId)

Determines whether the first specified [OpcNodeId](#) object is greater than the second specified [OpcNodeId](#) object.

C#

```
public static bool operator >(OpcNodeId left, OpcNodeId right)
```

GreaterThanOrEqual(OpcNodeId, OpcNodeId)

Determines whether the first specified [OpcNodeId](#) object is greater than or equal to the second specified [OpcNodeId](#) object.

C#

```
public static bool operator >=(OpcNodeId left, OpcNodeId right)
```

Implicit(ExpandedNodeId to OpcNodeId)

Converts a [ExpandedNodeId](#) to an [OpcNodeId](#) object.

C#

```
[CLSCompliant(false)]  
public static implicit operator OpcNodeId(ExpandedNodeId value)
```

Implicit(Int32 to OpcNodeId)

Converts a [Int32](#) to an [OpcNodeId](#) object.

C#

```
public static implicit operator OpcNodeId(int value)
```

Implicit(NodeId to OpcNodeId)

Converts a [NodeId](#) to an [OpcNodeId](#) object.

C#

```
[CLSCompliant(false)]  
public static implicit operator OpcNodeId(NodeId value)
```

Implicit(String to OpcNodeId)

Converts a [String](#) to an [OpcNodeId](#) object.

C#

```
public static implicit operator OpcNodeId(string value)
```

Inequality(OpcNodeId, OpcNodeId)

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

C#

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

LessThan(OpcNodeId, OpcNodeId)

Determines whether the first specified [OpcNodeId](#) object is less than the second specified [OpcNodeId](#) object.

C#

```
public static bool operator <(OpcNodeId left, OpcNodeId right)
```

LessThanOrEqual(OpcNodeId, OpcNodeId)

Determines whether the first specified [OpcNodeId](#) object is less than or equal to the second [OpcNodeId](#) object.

C#

```
public static bool operator <=(OpcNodeId left, OpcNodeId right)
```

Table of Contents

Constructors	1
OpcNodeId(Byte[])	1
OpcNodeId(Byte[], Int32)	1
OpcNodeId(Byte[], OpcNamespace)	1
OpcNodeId(Guid)	2
OpcNodeId(Guid, Int32)	2
OpcNodeId(Guid, OpcNamespace)	2
OpcNodeId(Int32)	3
OpcNodeId(Int32, Int32)	3
OpcNodeId(Int32, OpcNamespace)	3
OpcNodeId(OpcNodeId)	4
OpcNodeId(String)	4
OpcNodeId(String, Int32)	4
OpcNodeId(String, Int32, OpcName[])	5
OpcNodeId(String, Int32, OpcNamePath)	5
OpcNodeId(String, OpcName[])	6
OpcNodeId(String, OpcNamePath)	6
OpcNodeId(String, OpcNamespace)	6
OpcNodeId(String, OpcNamespace, OpcName[])	7
OpcNodeId(String, OpcNamespace, OpcNamePath)	7
OpcNodeId(UInt32)	8
OpcNodeId(UInt32, Int32)	8
OpcNodeId(UInt32, OpcNamespace)	8
Fields	9
DefaultSeparator	9
DefaultSeparatorOfPath	9
Properties	9
Factory	9
IsAbsolute	10
IsNull	10
Namespace	10
NamespaceIndex	11
NamespaceUri	11
Null	11
OriginalFormat	12
OriginalString	12
Path	12
Type	13
Value	13
ValueAsString	13
Methods	14
CompareTo(Object)	14
CompareTo(OpcNodeId)	14
Equals(Object)	14
Equals(OpcNodeId)	15
Equals(OpcNodeId, OpcNodeId)	15
Equals(OpcNodeId, OpcNodeId, OpcNodeIdComparison)	16
Equals(OpcNodeId, OpcNodeIdComparison)	16
ExcludePath()	17
GetHashCode()	17
IsNullOrEmpty(OpcNodeId)	17

Of(OpcName)	18
Of(OpcName, OpcNodeId)	19
Of(OpcName, OpcNodeId, Char)	19
Parse(String)	20
Resolve(IOpcNamespaceResolver)	21
ToString()	21
ToString(OpcNodeIdFormat)	21
TryParse(String, out OpcNodeId)	22
UriIdentifier(OpcNodeId)	22
WithPath(OpcName[])	23
WithPath(OpcNamePath)	23
Operators	24
Equality(OpcNodeId, OpcNodeId)	24
Explicit(OpcNodeId to ExpandedNodeId)	24
Explicit(OpcNodeId to NodeId)	24
GreaterThan(OpcNodeId, OpcNodeId)	25
GreaterThanOrEqual(OpcNodeId, OpcNodeId)	25
Implicit(ExpandedNodeId to OpcNodeId)	25
Implicit(Int32 to OpcNodeId)	25
Implicit(NodeId to OpcNodeId)	25
Implicit(String to OpcNodeId)	26
Inequality(OpcNodeId, OpcNodeId)	26
LessThan(OpcNodeId, OpcNodeId)	26
LessThanOrEqual(OpcNodeId, OpcNodeId)	26