

OpcNamespace Class

Namespace: Opc.UaFx

Assemblies: Opc.UaFx.Advanced.dll

Represents a nominal space used to categorize information like nodes, types, identifiers, etc. under a logical topic.

C#

```
public class OpcNamespace : IEquatable<OpcNamespace>, IComparable, IComparable<OpcNamespace>
```

Inheritance [Object](#) › [OpcNamespace](#)

Implements [IEquatable<OpcNamespace>](#), [IComparable](#), [IComparable<OpcNamespace>](#)

Remarks

An OPC UA server provides an array of such topics known as namespaces. Within these list of namespaces a dedicated namespace can be determined by its unique [Index](#). Its [Uri](#) does additionally inform about the general purpose, the source or the owner of the namespace.

The [Default](#) namespace uses an [Index](#) equals zero and a null reference (Nothing in Visual Basic) as the [Uri](#). A system or custom defined [OpcNamespace](#) instance can have either an [Index](#) unequal to zero and/or an [Uri](#) unequal to a null reference (Nothing in Visual Basic).

A [OpcNamespace](#) can be constructed from a simple [Uri](#). In case there the [Host](#) portion uses the [UriHost](#) the whole [Uri](#) information is not used for the [Uri](#) of the [OpcNamespace](#) except the [Port](#) portion. If it is not the scheme specific default port (see [IsDefaultPort](#)) and it is greater than zero it is used as the [Index](#) of the namespace.

In case there the [OpcNamespace](#) is constructed from a [Uri](#) which uses a different [Host](#) portion than [UriHost](#) the whole [Uri](#) is used as the [Uri](#) of the namespace except of an existing [Port](#) portion. If the [Port](#) portion is greater than zero and is not the scheme specific default port (see [IsDefaultPort](#)) it is used as the [Index](#) of the namespace. If the [Scheme](#) is equals to [UriSchemeUriSchemeHttp](#) is used instead for the [Uri](#) of the namespace. Additionally any [Fragment](#) and [Query](#) component is also discarded and any trailing '/' is trimmed from [Path](#) component of the [Uri](#).

Fields

Name	Description
UriHost	Specifies that an Uri is a pointer to a OPC UA namespace without a specific namespace Host .
UriScheme	Specifies that an Uri is a pointer to a OPC UA namespace without a specific namespace Scheme .

Properties

Name	Description
Default	Gets the default namespace used in cases there the concrete does not matter or is just unknown.
Index	Gets the index of the namespace within all namespaces of a server.
IsAbsolute	Gets a value indicating whether the namespace is absolute.
IsDefault	Gets a value indicating whether the namespace defines the default namespace without to refer to a specific address space.
IsResolved	Gets a value indicating whether the namespace is resolved regarding its Index , Uri and Value .
Scope	Gets scope within the current OpcNamespace has been declared and its metadata applies to.
Uri	Gets the uniform resource identifier (URI) of the namespace represented and referred to by the Index .
Value	Gets the intrinsic value of the namespace represented and referred to by the Index .

Methods

Name	Description
Clear	Empties the known namespaces.
CompareTo(Object)	Compares the current OpcNamespace with the other .
CompareTo(OpcNamespace)	Compares the current OpcNamespace with another OpcNamespace .
Create(String)	Creates a OpcNamespace using the value specified.
Create(String, Int32)	Creates a OpcNamespace using the value and index specified.
Create(Uri)	Creates a OpcNamespace from the Uri specified. If uri provides Port information, the Port is used for the Index of the OpcNamespace created.
Create(Uri, Int32)	Creates a OpcNamespace from the Uri specified. If uri provides Port information, the Port is used for the Index of the OpcNamespace created, if index not greater than zero.
Equals(Object)	Determines whether the specified other is equal to this OpcNamespace .
Equals(OpcNamespace)	Determines whether the specified other is equal to this OpcNamespace .
Get(Int32)	Retrieves the OpcNamespace known under the namespaceIndex specified.
Get(Int32, String)	Retrieves the OpcNamespace known under the namespaceIndex and namespaceValue specified.
Get(Int32, Uri)	Retrieves the OpcNamespace known under the namespaceIndex and namespaceUri specified.
Get(String)	Retrieves the OpcNamespace using the information provided by the namespaceUriOrValue specified.
Get(Uri)	Retrieves the OpcNamespace with the information provided by the namespaceUri specified.
GetHashCode	Retrieves a hash code for this OpcNamespace .
GetId(Byte)	Retrieves a new OpcNodeId using the opaque value and this OpcNamespace .
GetId(Guid)	Retrieves a new OpcNodeId using the general unique identifier (= GUID) value and this OpcNamespace .
GetId(Int32)	Retrieves a new OpcNodeId using the numeric value and this OpcNamespace .

Name	Description
GetId(Object)	Retrieves a new OpcNodeId using the value and this OpcNamespace .
GetId(String)	Retrieves a new OpcNodeId using the Stringvalue and this OpcNamespace .
GetId(String, OpcName)	Retrieves a new OpcNodeId using the Stringvalue , this OpcNamespace and the pathElements .
GetId(String, OpcNamePath)	Retrieves a new OpcNodeId using the Stringvalue , this OpcNamespace and the path .
GetId(UInt32)	Retrieves a new OpcNodeId using the numeric value and this OpcNamespace .
GetName(String)	Retrieves a new OpcName using the name and this OpcNamespace .
Parse(String)	Converts a namespace string to a OpcNamespace instance.
Resolve(IOpcNamespaceResolver)	Resolves missing OPC UA namespace information using the object specified by resolver .
ToString	Returns a string representing the namespace.
TryCreate(Uri, Int32, OpcNamespace@)	Determines whether a Uri is a valid namespace.
TryCreate(Uri, OpcNamespace@)	Determines whether a Uri is a valid namespace.
TryParse(String, OpcNamespace@)	Determines whether a string is a valid namespace.

Operators

Name	Description
op_Equality(OpcNamespace, OpcNamespace)	Returns a value indicating whether two instance of OpcNamespace are equal.
op_GreaterThan(OpcNamespace, OpcNamespace)	Determines whether the first specified OpcNamespace object is greater than the second specified OpcNamespace object.
op_GreaterThanOrEqual(OpcNamespace, OpcNamespace)	Determines whether the first specified OpcNamespace object is greater than or equal to the second specified OpcNamespace object.
op_Implicit(UaFx.OpcNamespace)	Converts a Int32 to an OpcNamespace object.
op_Implicit(String)	Converts a String to an OpcNamespace object.
op_Inequality(OpcNamespace, OpcNamespace)	Returns a value indicating whether two instances of OpcNamespace are not equal.
op_LessThan(OpcNamespace, OpcNamespace)	Determines whether the first specified OpcNamespace object is less than the second specified OpcNamespace object.
op_LessThanOrEqual(OpcNamespace, OpcNamespace)	Determines whether the first specified OpcNamespace object is less than or equal to the second OpcNamespace object.

Table of Contents

Remarks	1
Fields	1
Properties	2
Methods	2
Operators	3