OpcNodeReferenceCollection Members

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

Constructors

OpcNodeReferenceCollection()

Initializes a new instance of the OpcNodeReferenceCollection class.

C#

```
public OpcNodeReferenceCollection()
```

Methods

Add(IOpcNode, IEnumerable<OpcReference>)

Adds a OpcNodeReference instance representing a node / references bundle using the `node` and the `references` specified.

C#

```
public OpcNodeReference Add(IOpcNode node, IEnumerable<OpcReference> references)
```

Parameters

`node` IOpcNode

The IOpcNode used as the node to which the `references` specified belong to.

`references` IEnumerable<OpcReference>

A sequence of OpcReference instances to associate with the node refered to by `node`.

Returns

OpcNodeReference

A new instance of the OpcNodeReference class which combines the `node` and the `references` specified.

Exceptions

ArgumentNullException

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

Add(IOpcNode, OpcNodeId)

Adds OpcNodeReference instances representing a bidirectional relation between the `sourceNode` and the node identified by the `targetNodeId` specified. The references created represent a ParentToChild and ChildToParent relation using Organizes as the type of reference.

C#

```
public OpcNodeReference[] Add(IOpcNode sourceNode, OpcNodeId targetNodeId)
```

Parameters

`sourceNode` IOpcNode

The source portion of the bidirectional reference defined between the passed IOpcNode and the target identified by the `targetNodeId`.

`targetNodeId` OpcNodeId

The OpcNodeId which identifies the target one reference will cease in, while the second reference uses it the other way round.

Returns

OpcNodeReference[]

An array of OpcNodeReference instances representing the references added to the collection to define the bidirectional relation between the `sourceNode` and the target identified by the `targetNodeId`.

Exceptions

ArgumentNullException

The `sourceNode` or `targetNodeId` is a null reference (Nothing in Visual Basic).

Add(IOpcNode, OpcNodeId, Nullable<OpcReferenceDirection>)

Adds OpcNodeReference instances representing either a bidirectional or a unidirectional relation between the `sourceNode` and the target identified by the `targetNodeId` specified. The references created depend on the value specified by `direction` using Organizes as the type of reference.

C#

```
public OpcNodeReference[] Add(IOpcNode sourceNode, OpcNodeId targetNodeId,
OpcReferenceDirection? direction)
```

Parameters

`sourceNode` IOpcNode

The source portion of the references defined between the passed IOpcNode and target identified by the `targetNodeId`.

`targetNodeId` OpcNodeId

The OpcNodeId which identifies the target one reference will cease in, while the second reference uses it the other way round; if `direction` is a null reference (Nothing in Visual Basic).

`direction` Nullable<OpcReferenceDirection>

One of the members defined by the OpcReferenceDirection enumeration which defines the kind of unidirectional relation the reference will construct or a null reference (Nothing in Visual Basic) if bidirectional references shall be defined.

Returns

OpcNodeReference[]

An array of OpcNodeReference instances representing the references addded to the collection to define the bidirectional or unidirectional relation between the `sourceNode` and the target identified by the `targetNodeId`.

Exceptions

ArgumentNullException

The `sourceNode` or `targetNodeId` is a null reference (Nothing in Visual Basic).

Add(IOpcNode, OpcNodeId, Nullable<OpcReferenceDirection>, OpcNodeId)

Adds OpcNodeReference instances representing either a bidirectional or a unidirectional relation between the `sourceNode` and the target identified by the `targetNodeId` specified. The references created depend on the value specified by `direction` using the type of reference identified by the `referenceTypeId` specified.

C#

```
public OpcNodeReference[] Add(IOpcNode sourceNode, OpcNodeId targetNodeId,
OpcReferenceDirection? direction, OpcNodeId referenceTypeId)
```

Parameters

`sourceNode` IOpcNode

The source portion of the references defined between the passed IOpcNode and target identified by the `targetNodeId`.

`targetNodeId` OpcNodeId

The OpcNodeId which identifies the target one reference will cease in, while the second reference uses it the other way round; if `direction` is a null reference (Nothing in Visual Basic).

`direction` Nullable<OpcReferenceDirection>

One of the members defined by the OpcReferenceDirection enumeration which defines the kind of unidirectional relation the reference will construct or a null reference (Nothing in Visual Basic) if bidirectional references shall be defined.

`referenceTypeId` OpcNodeId

The OpcNodeId which identifies the type of reference to represent between the target identified by the `targetNodeId` and its opponent.

Returns

OpcNodeReference[]

An array of OpcNodeReference instances representing the references addded to the collection to define the bidirectional or unidirectional relation between the `sourceNode` and the target identified by the `targetNodeId`.

Exceptions

ArgumentNullException

The `sourceNode` or `targetNodeId` is a null reference (Nothing in Visual Basic).

Add(IOpcNode, OpcNodeId, Nullable<OpcReferenceDirection>, OpcReferenceType)

Adds OpcNodeReference instances representing either a bidirectional or a unidirectional relation between the `sourceNode` and the target identified by the `targetNodeId` specified. The references created depend on the value specified by `direction` using the `referenceType` as the type of reference.

C#

```
public OpcNodeReference[] Add(IOpcNode sourceNode, OpcNodeId targetNodeId,
OpcReferenceDirection? direction, OpcReferenceType referenceType)
```

Parameters

`sourceNode` IOpcNode

The source portion of the references defined between the passed IOpcNode and target identified by the `targetNodeId`.

`targetNodeId` OpcNodeId

The OpcNodeId which identifies the target one reference will cease in, while the second reference uses it the other way round; if `direction` is a null reference (Nothing in Visual Basic).

`direction` Nullable<OpcReferenceDirection>

One of the members defined by the OpcReferenceDirection enumeration which defines the kind of unidirectional relation the reference will construct or a null reference (Nothing in Visual Basic) if bidirectional references shall be defined.

`referenceType` OpcReferenceType

One of the members defined by the OpcReferenceType enumeration to use to setup the type of reference represented.

Returns

OpcNodeReference[]

An array of OpcNodeReference instances representing the references addded to the collection to define the bidirectional or unidirectional relation between the `sourceNode` and the target identified by the `targetNodeId`.

Exceptions

ArgumentNullException

The `sourceNode` or `targetNodeId` is a null reference (Nothing in Visual Basic).

Add(IOpcNode, OpcReference[])

Adds a OpcNodeReference instance representing a node / references bundle using the `node` and the `references` specified.

C#

```
public OpcNodeReference Add(IOpcNode node, params OpcReference[] references)
```

Parameters

`node` IOpcNode

The IOpcNode used as the node to which the `references` specified belong to.

`references` OpcReference[]

An array of OpcReference instances to associate with the node refered to by `node`.

Returns

OpcNodeReference

A new instance of the OpcNodeReference class which combines the `node` and the `references` specified.

Exceptions

ArgumentNullException

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

Add(OpcNodeId, IEnumerable<OpcReference>)

Adds a OpcNodeReference instance representing a node / references bundle using the node identified by the `nodeId` and the `references` specified.

C#

```
public OpcNodeReference Add(OpcNodeId nodeId, IEnumerable<OpcReference> references)
```

Parameters

`nodeId` OpcNodeId

The OpcNodeId which identifies the node to which the `references` specified belong to.

`references` IEnumerable<OpcReference>

A sequence of OpcReference instances to associate with the node identified by the `nodeId`.

Returns

OpcNodeReference

A new instance of the OpcNodeReference class which combines the `nodeId` and the `references` specified.

Exceptions

ArgumentNullException

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

Add(OpcNodeId, OpcNodeId)

Adds OpcNodeReference instances representing a bidirectional relation between the source identified by the `sourceNodeId` and the node identified by the `targetNodeId` specified. The references created represent a ParentToChild and ChildToParent relation using Organizes as the type of reference.

C#

```
public OpcNodeReference[] Add(OpcNodeId sourceNodeId, OpcNodeId targetNodeId)
```

Parameters

`sourceNodeId` OpcNodeId

The OpcNodeId which identifies the source portion of the bidirectional reference defined between the passed IOpcNode and the target identified by the `targetNodeId`.

`targetNodeId` OpcNodeId

The OpcNodeId which identifies the target one reference will cease in, while the second reference uses it the other way round.

Returns

OpcNodeReference[]

An array of OpcNodeReference instances representing the references added to the collection to define the bidirectional relation between the source identified by the `sourceNodeId` and the target identified by the `targetNodeId`.

Exceptions

ArgumentNullException

The `sourceNodeId` or `targetNodeId` is a null reference (Nothing in Visual Basic).

Add(OpcNodeId, OpcNodeId, Nullable<OpcReferenceDirection>)

Adds OpcNodeReference instances representing either a bidirectional or a unidirectional relation between the source identified by `sourceNodeId` and the target identified by the `targetNodeId` specified. The references created depend on the value specified by `direction` using Organizes as the type of reference.

C#

```
public OpcNodeReference[] Add(OpcNodeId sourceNodeId, OpcNodeId targetNodeId,
OpcReferenceDirection? direction)
```

Parameters

`sourceNodeId` OpcNodeId

The OpcNodeId which identifies the source portion of the references defined between the passed IOpcNode and target identified by the `targetNodeId`.

`targetNodeId` OpcNodeId

The OpcNodeId which identifies the target one reference will cease in, while the second reference uses it the other way round; if `direction` is a null reference (Nothing in Visual Basic).

`direction` Nullable<OpcReferenceDirection>

One of the members defined by the OpcReferenceDirection enumeration which defines the kind of unidirectional relation the reference will construct or a null reference (Nothing in Visual Basic) if bidirectional references shall be defined.

Returns

OpcNodeReference[]

An array of OpcNodeReference instances representing the references addded to the collection to define

the bidirectional or unidirectional relation between the source identified by the `sourceNodeId` and the target identified by the `targetNodeId`.

Exceptions

ArgumentNullException

The `sourceNodeId` or `targetNodeId` is a null reference (Nothing in Visual Basic).

Add(OpcNodeId, OpcNodeId, Nullable<OpcReferenceDirection>, OpcNodeId)

Adds OpcNodeReference instances representing either a bidirectional or a unidirectional relation between the source identified by the `sourceNodeId` and the target identified by the `targetNodeId` specified. The references created depend on the value specified by `direction` using the type of reference identified by the `referenceTypeId` specified.

C#

```
public OpcNodeReference[] Add(OpcNodeId sourceNodeId, OpcNodeId targetNodeId,
OpcReferenceDirection? direction, OpcNodeId referenceTypeId)
```

Parameters

`sourceNodeId` OpcNodeId

The OpcNodeId which identifies the source portion of the references defined between the passed IOpcNode and target identified by the `targetNodeId`.

`targetNodeId` OpcNodeId

The OpcNodeId which identifies the target one reference will cease in, while the second reference uses it the other way round; if `direction` is a null reference (Nothing in Visual Basic).

`direction` Nullable<OpcReferenceDirection>

One of the members defined by the OpcReferenceDirection enumeration which defines the kind of unidirectional relation the reference will construct or a null reference (Nothing in Visual Basic) if bidirectional references shall be defined.

`referenceTypeId` OpcNodeId

The OpcNodeId which identifies the type of reference to represent between the target identified by the `targetNodeId` and its opponent.

Returns

OpcNodeReference[]

An array of OpcNodeReference instances representing the references addded to the collection to define the bidirectional or unidirectional relation between the source identified by the `sourceNodeId` and the target identified by the `targetNodeId`.

Exceptions

ArgumentNullException

The `sourceNodeId` or `targetNodeId` is a null reference (Nothing in Visual Basic).

Add(OpcNodeId, OpcNodeId, Nullable<OpcReferenceDirection>, OpcReferenceType)

Adds OpcNodeReference instances representing either a bidirectional or a unidirectional relation between the source identified by the `sourceNodeId` and the target identified by the `targetNodeId` specified. The references created depend on the value specified by `direction` using the `referenceType` as the type of reference.

C#

```
public OpcNodeReference[] Add(OpcNodeId sourceNodeId, OpcNodeId targetNodeId,
OpcReferenceDirection? direction, OpcReferenceType referenceType)
```

Parameters

`sourceNodeId` OpcNodeId

The OpcNodeId which identifies the source portion of the references defined between the passed IOpcNode and target identified by the `targetNodeId`.

`targetNodeId` OpcNodeId

The OpcNodeId which identifies the target one reference will cease in, while the second reference uses it the other way round; if `direction` is a null reference (Nothing in Visual Basic).

`direction` Nullable<OpcReferenceDirection>

One of the members defined by the OpcReferenceDirection enumeration which defines the kind of unidirectional relation the reference will construct or a null reference (Nothing in Visual Basic) if bidirectional references shall be defined.

`referenceType` OpcReferenceType

One of the members defined by the OpcReferenceType enumeration to use to setup the type of reference represented.

Returns

OpcNodeReference[]

An array of OpcNodeReference instances representing the references addded to the collection to define the bidirectional or unidirectional relation between the source identified by the `sourceNodeId` and the target identified by the `targetNodeId`.

Exceptions

ArgumentNullException

The `sourceNodeId` or `targetNodeId` is a null reference (Nothing in Visual Basic).

Add(OpcNodeId, OpcReference[])

Adds a OpcNodeReference instance representing a node / references bundle using the node identified by the `nodeId` and the `references` specified.

C#

```
public OpcNodeReference Add(OpcNodeId nodeId, params OpcReference[] references)
```

Parameters

`nodeId` OpcNodeId

The OpcNodeId which identifies the node to which the `references` specified belong to.

`references` OpcReference[]

An array of OpcReference instances to associate with the node identified by the `nodeId`.

Returns

OpcNodeReference

A new instance of the OpcNodeReference class which combines the `nodeId` and the `references` specified.

Exceptions

ArgumentNullException

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

GetKeyForItem(OpcNodeReference)

Extracts the key from the specified element.

C#

```
protected override OpcNodeId GetKeyForItem(OpcNodeReference item)
```

Parameters

`item` OpcNodeReference

The `item` from which to extract the key.

Returns

OpcNodeId

The key for the specified `item`.

Exceptions

ArgumentNullException

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

Table of Contents