OpcNodeManager Members

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

Constructors

OpcNodeManager(OpcNamespace, OpcNamespace[])

Initializes a new instance of the OpcNodeManager class using the `defaultNamespace` and additionally specified `namespaces`.

C#

```
protected OpcNodeManager(OpcNamespace defaultNamespace, params OpcNamespace[] namespaces)
```

Parameters

`defaultNamespace` OpcNamespace

The default OpcNamespace used for the DefaultNamespace property. In case there AssignDefaultNamespaceOfManager is equals true (the default value) the Name and Id of the OpcNode instances defined are linked to this namespace in case there is no namespace information associated to them.

`namespaces` OpcNamespace[]

An array of OpcNamespace instances indicating which namespaces are as well maintained by the new OpcNodeManager. The instances can then found in the Namespaces property as well.

Exceptions

ArgumentException

The `defaultNamespace` or one of the items in `namespaces` refers to a namespace which presets a namespace index. The according namespace index is determined by the server.

ArgumentNullException

The `defaultNamespace` or one of the items in `namespaces` is a null reference (Nothing in Visual Basic).

OpcNodeManager(String, String[])

Initializes a new instance of the OpcNodeManager class using the `defaultNamespace` and additionally specified String-based `namespaces`.

C#

```
protected OpcNodeManager(string defaultNamespace, params string[] namespaces)
```

Parameters

`defaultNamespace` String

The String representation of the default OpcNamespace which is used for the DefaultNamespace property. In case there AssignDefaultNamespaceOfManager is equals true (the default value) the Name and Id of the OpcNode instances defined are linked to this namespace in case there is no namespace information associated to them.

`namespaces` String[]

An array of String representations of the OpcNamespace indicating which namespaces are as well maintained by the new OpcNodeManager. The used OpcNamespace instances can be found in the Namespaces property as well.

Exceptions

ArgumentException

The `defaultNamespace` or one of the items in `namespaces` is an empty string.

ArgumentNullException

The `defaultNamespace` or one of the items in `namespaces` is a null reference (Nothing in Visual Basic).

OpcNodeManager(Uri, Uri[])

Initializes a new instance of the OpcNodeManager class using the `defaultNamespaceUri` and additionally specified Uri-based `namespaceUris`.

C#

```
protected OpcNodeManager(Uri defaultNamespaceUri, params Uri[] namespaceUris)
```

Parameters

`defaultNamespaceUri` Uri

The Uri representation of the default OpcNamespace which is used for the DefaultNamespace property. In case there AssignDefaultNamespaceOfManager is equals true (the default value) the Name and Id of the OpcNode instances defined are linked to this namespace in case there is no namespace information associated to them.

`namespaceUris` Uri[]

An array of Uri representations of the OpcNamespace indicating which namespaces are as well maintained by the new OpcNodeManager. The used OpcNamespace instances can be found in the Namespaces property as well.

Exceptions

ArgumentNullException

The `defaultNamespaceUri` or one of the items in `namespaceUris` is a null reference (Nothing in Visual

Basic).

Events

AfterRead

C#

```
public event OpcNodeAccessEventHandler AfterRead
```

AfterWrite

C#

```
public event OpcNodeAccessEventHandler AfterWrite
```

BeforeRead

C#

```
public event OpcNodeAccessEventHandler BeforeRead
```

BeforeWrite

C#

```
public event OpcNodeAccessEventHandler BeforeWrite
```

MonitoredItemCreated

Occurs when a new monitored item is created through a OpcSession.

C#

```
public event OpcMonitoredItemEventHandler MonitoredItemCreated
```

MonitoredItemDeleted

Occurs when an existing monitored item of a OpcSession has been deleted.

C#

```
public event OpcMonitoredItemEventHandler MonitoredItemDeleted
```

MonitoredItemModified

Occurs when an existing monitored item of a OpcSession has been modified.

C#

```
public event OpcMonitoredItemEventHandler MonitoredItemModified
```

MonitoredItemsCreated

Occurs when a set of new monitored items is created through a OpcSession.

C#

```
public event OpcMonitoredItemsEventHandler MonitoredItemsCreated
```

MonitoredItemsDeleted

Occurs when a set of existing monitored items of a OpcSession has been deleted.

C#

```
public event OpcMonitoredItemsEventHandler MonitoredItemsDeleted
```

MonitoredItemsModified

Occurs when a set of existing monitored items of a OpcSession has been modified.

C#

```
public event OpcMonitoredItemsEventHandler MonitoredItemsModified
```

Properties

DefaultNamespace

C#

```
public OpcNamespace DefaultNamespace { get; }
```

Property Value

OpcNamespace

DefaultNamespaceIndex

C#

```
public int DefaultNamespaceIndex { get; }
```

Property Value

Int32

DefaultNamespaceUri

C#

```
public Uri DefaultNamespaceUri { get; }
```

Property Value

Uri

DefaultNamespaceValue

C#

```
public string DefaultNamespaceValue { get; }
```

Property Value

String

Namespaces

C#

```
public OpcReadOnlyNamespaceCollection Namespaces { get; }
```

Property Value

OpcReadOnlyNamespaceCollection

NodeIdFactory

Gets or sets the OpcNodeIdFactory instance used by this node manager to determine / create node identifiers for the nodes provided / defined by the node manager.

C#

```
protected virtual OpcNodeIdFactory NodeIdFactory { get; set; }
```

Property Value

OpcNodeIdFactory

An instance of the OpcNodeIdFactory class. In case there is no custom OpcNodeIdFactory defined. The Factory is used.

Nodes

C#

```
public OpcReadOnlyNodeCollection Nodes { get; }
```

Property Value

OpcReadOnlyNodeCollection

NotifierNodes

C#

```
public OpcReadOnlyNodeCollection NotifierNodes { get; }
```

Property Value

OpcReadOnlyNodeCollection

SyncRoot

C#

```
protected object SyncRoot { get; }
```

Property Value

Object

SystemContext

C#

```
protected OpcContext SystemContext { get; }
```

Property Value

OpcContext

Methods

AddNode(IOpcNode)

C#

```
protected void AddNode(IOpcNode node)
```

Parameters

`node` IOpcNode

Exceptions

ArgumentNullException

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

AddNode(IOpcNode, IEnumerable<IOpcNodeReferenceAware>)

C#

```
protected void AddNode(IOpcNode node, IEnumerable<IOpcNodeReferenceAware> references)
```

Parameters

`node` IOpcNode

`references` IEnumerable<IOpcNodeReferenceAware>

Exceptions

ArgumentNullException

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

AddNode(IOpcNode, IOpcNodeReferenceAware[])

C#

```
protected void AddNode(IOpcNode node, params IOpcNodeReferenceAware[] references)
```

Parameters

`node` IOpcNode

`references` IOpcNodeReferenceAware[]

Exceptions

ArgumentNullException

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

AddNode(IOpcNode, OpcNodeReferenceCollection)

C#

```
protected void AddNode(IOpcNode node, OpcNodeReferenceCollection references)
```

Parameters

`node` IOpcNode

`references` OpcNodeReferenceCollection

Exceptions

ArgumentNullException

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

AddNode(OpcContext, IOpcNode)

C#

```
protected void AddNode(OpcContext context, IOpcNode node)
```

Parameters

`context` OpcContext

`node` IOpcNode

Exceptions

ArgumentNullException

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

AddNode(OpcContext, IOpcNode, IEnumerable<IOpcNodeReferenceAware>)

C#

```
protected virtual void AddNode(OpcContext context, IOpcNode node,
IEnumerable<IOpcNodeReferenceAware> references)
```

Parameters

`context` OpcContext

`node` IOpcNode

`references` IEnumerable<IOpcNodeReferenceAware>

Exceptions

ArgumentNullException

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

AddNode(OpcContext, IOpcNode, IOpcNodeReferenceAware[])

C#

```
protected void AddNode(OpcContext context, IOpcNode node, params IOpcNodeReferenceAware[]
references)
```

Parameters

`context` OpcContext

`node` IOpcNode

`references` IOpcNodeReferenceAware[]

Exceptions

ArgumentNullException

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

AddNode(OpcContext, IOpcNode, OpcNodeReferenceCollection)

C#

```
protected virtual void AddNode(OpcContext context, IOpcNode node, OpcNodeReferenceCollection
references)
```

Parameters

`context` OpcContext

`node` IOpcNode

`references` OpcNodeReferenceCollection

Exceptions

ArgumentNullException

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

AddNotifierNode(IOpcNode)

C#

```
protected void AddNotifierNode(IOpcNode node)
```

Parameters

`node` IOpcNode

Exceptions

ArgumentNullException

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

Browse(OpcContext, OpcBrowsePoint, OpcReferenceDescriptionCollection)

C#

```
protected virtual OpcBrowsePoint Browse(OpcContext context, OpcBrowsePoint browsePoint,
OpcReferenceDescriptionCollection references)
```

Parameters

`context` OpcContext

`browsePoint` OpcBrowsePoint

`references` OpcReferenceDescriptionCollection

Returns

OpcBrowsePoint

CreateHistory(IOpcNode, OpcHistoryModificationInfo, OpcValueCollection)

C#

```
protected virtual OpcStatusCollection CreateHistory(IOpcNode node,
OpcHistoryModificationInfo modificationInfo, OpcValueCollection values)
```

Parameters

`node` IOpcNode

`modificationInfo` OpcHistoryModificationInfo

`values` OpcValueCollection

Returns

OpcStatusCollection

CreateNodes(OpcNodeReferenceCollection)

C#

```
protected virtual IEnumerable<IOpcNode> CreateNodes(OpcNodeReferenceCollection references)
```

Parameters

`references` OpcNodeReferenceCollection

Returns

IEnumerable<IOpcNode>

DeleteHistory(IOpcNode, OpcHistoryModificationInfo, IEnumerable<DateTime>)

C#

```
protected virtual OpcStatusCollection DeleteHistory(IOpcNode node,
OpcHistoryModificationInfo modificationInfo, IEnumerable<DateTime> times)
```

Parameters

`node` IOpcNode

`modificationInfo` OpcHistoryModificationInfo

`times` IEnumerable<DateTime>

Returns

OpcStatusCollection

DeleteHistory(IOpcNode, OpcHistoryModificationInfo, Nullable<DateTime>, Nullable<DateTime>, OpcDeleteHistoryOptions)

C#

```
protected virtual OpcStatusCollection DeleteHistory(IOpcNode node,
OpcHistoryModificationInfo modificationInfo, DateTime? startTime, DateTime? endTime,
OpcDeleteHistoryOptions options)
```

Parameters

`node` IOpcNode

`modificationInfo` OpcHistoryModificationInfo

`startTime` Nullable<DateTime>

`endTime` Nullable<DateTime>

`options` OpcDeleteHistoryOptions

Returns

OpcStatusCollection

DeleteHistory(IOpcNode, OpcHistoryModificationInfo, OpcValueCollection)

C#

```
protected virtual OpcStatusCollection DeleteHistory(IOpcNode node,
OpcHistoryModificationInfo modificationInfo, OpcValueCollection values)
```

Parameters

`node` IOpcNode

`modificationInfo` OpcHistoryModificationInfo

`values` OpcValueCollection

Returns

OpcStatusCollection

DeleteNodes()

C#

```
protected virtual void DeleteNodes()
```

Dispose()

Releases all resources used by the OpcNodeManager.

C#

```
public void Dispose()
```

Dispose(Boolean)

Releases the unmanaged resources used by the OpcNodeManager and optionally releases the managed resources.

C#

```
protected virtual void Dispose(bool disposing)
```

Parameters

`disposing` Boolean

The value true to release both managed and unmanaged resources; otherwise the value false to release only unmanaged resources.

ExportNodes()

C#

```
public IEnumerable<OpcNodeSet> ExportNodes()
```

Returns

IEnumerable<OpcNodeSet>

ExportNodes(OpcNodeExportOptions)

C#

```
public IEnumerable<OpcNodeSet> ExportNodes(OpcNodeExportOptions options)
```

Parameters

`options` OpcNodeExportOptions

Returns

IEnumerable<OpcNodeSet>

GetNode(OpcNodeId)

Retrieves the IOpcNode which is maintained by the Nodes of this OpcNodeManager and is known under the `nodeId` specified.

C#

```
public IOpcNode GetNode(OpcNodeId nodeId)
```

Parameters

`nodeId` OpcNodeId

The OpcNodeId of the IOpcNode instance to determine.

Returns

IOpcNode

An instance implementing the IOpcNode interface its Id attribute is equals to the `nodeId` specified if found; otherwise a null reference (Nothing in Visual Basic).

Exceptions

ArgumentNullException

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

GetNode(String)

Retrieves the IOpcNode which is maintained by the Nodes of this OpcNodeManager and is known under the `nodePath` specified.

C#

```
public IOpcNode GetNode(string nodePath)
```

Parameters

`nodePath` String

A sequence of browse and symbolic names combined using a slash ('/') to describe the 'path' to the node to determine. To query a node using its browse name use the namespace qualified name format 'namespaceIndex:name'; otherwise the 'part of the path' is compared to the symbolic name of the node.

Returns

IOpcNode

An instance implementing the IOpcNode interface its Name or SymbolicName matches the last 'part' in the `nodePath` specified, if found; otherwise a null reference (Nothing in Visual Basic).

Exceptions

ArgumentException

The `nodePath` is an empty string.

ArgumentNullException

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

Remarks

The first 'part' in the `nodePath` shall refer to a unique browse or symbolic name of a node maintained by the Nodes; otherwise the first node which matches the browse or symbolic name specified is used to continue the node query. Note that this method does not support querying nodes which are organized in an ambiguous way (according to their sub-tree they reside).

ImplementNode(IOpcNode)

When implemented in a derived class, implements the logic required to integrate the `node` into the underlying system.

C#

```
protected virtual void ImplementNode(IOpcNode node)
```

Parameters

`node` IOpcNode

An instance implementing the IOpcNode interface which is to be integrated into the underlying system.

Remarks

This method is called during the whole address space creation. This includes the import of nodes using ImportNodes and the creation of nodes using CreateNodes(OpcNodeReferenceCollection). Each node imported / created is passed to this method to integrate it accordingly into the underlying system.

Additionally this method is called after the startup of the server whenever a client adds another node to the address space part(s) maintained by the current OpcNodeManager.

ImportNodes()

When implemented in a derived class, offers the different OpcNodeSets which are imported prior any other nodes are created within the address space part organized by the current OpcNodeManager.

C#

```
protected virtual IEnumerable<OpcNodeSet> ImportNodes()
```

Returns

IEnumerable<OpcNodeSet>

A sequence of OpcNodeSet instances to use to import external stored node information into the address space.

IsNodeAccessible(OpcContext, OpcNodeId, IOpcNodeInfo)

C#

```
protected virtual bool IsNodeAccessible(OpcContext context, OpcNodeId viewId, IOpcNodeInfo
node)
```

Parameters

`context` OpcContext

`viewId` OpcNodeId

`node` IOpcNodeInfo

Returns

Boolean

Exceptions

ArgumentNullException

OnAfterRead(OpcNodeAccessEventArgs)

C#

```
protected virtual void OnAfterRead(OpcNodeAccessEventArgs e)
```

Parameters

`e` OpcNodeAccessEventArgs

OnAfterWrite(OpcNodeAccessEventArgs)

C#

```
protected virtual void OnAfterWrite(OpcNodeAccessEventArgs e)
```

Parameters

e OpcNodeAccessEventArgs

OnBeforeRead(OpcNodeAccessEventArgs)

C#

```
protected virtual void OnBeforeRead(OpcNodeAccessEventArgs e)
```

Parameters

e OpcNodeAccessEventArgs

OnBeforeWrite(OpcNodeAccessEventArgs)

C#

```
protected virtual void OnBeforeWrite(OpcNodeAccessEventArgs e)
```

Parameters

e OpcNodeAccessEventArgs

OnMonitoredItemCreated(OpcMonitoredItemEventArgs)

Raises the MonitoredItemCreated event of the OpcNodeManager.

C#

```
protected virtual void OnMonitoredItemCreated(OpcMonitoredItemEventArgs e)
```

Parameters

e OpcMonitoredItemEventArgs

The event data.

OnMonitoredItemDeleted(OpcMonitoredItemEventArgs)

Raises the MonitoredItemDeleted event of the OpcNodeManager.

C#

```
protected virtual void OnMonitoredItemDeleted(OpcMonitoredItemEventArgs e)
```

Parameters

e OpcMonitoredItemEventArgs

The event data.

OnMonitoredItemModified(OpcMonitoredItemEventArgs)

Raises the MonitoredItemModified event of the OpcNodeManager.

C#

```
protected virtual void OnMonitoredItemModified(OpcMonitoredItemEventArgs e)
```

Parameters

e OpcMonitoredItemEventArgs

The event data.

OnMonitoredItemsCreated(OpcMonitoredItemsEventArgs)

Raises the MonitoredItemsCreated event of the OpcNodeManager.

C#

```
protected virtual void OnMonitoredItemsCreated(OpcMonitoredItemsEventArgs e)
```

Parameters

e OpcMonitoredItemsEventArgs

The event data.

OnMonitoredItemsDeleted(OpcMonitoredItemsEventArgs)

Raises the MonitoredItemsDeleted event of the OpcNodeManager.

C#

```
protected virtual void OnMonitoredItemsDeleted(OpcMonitoredItemsEventArgs e)
```

Parameters

e OpcMonitoredItemsEventArgs

The event data.

OnMonitoredItemsModified(OpcMonitoredItemsEventArgs

)

Raises the MonitoredItemsModified event of the OpcNodeManager.

C#

```
protected virtual void OnMonitoredItemsModified(OpcMonitoredItemsEventArgs e)
```

Parameters

`e` OpcMonitoredItemsEventArgs

The event data.

QueryNode(OpcNodeId)

C#

```
protected virtual IOpcNode QueryNode(OpcNodeId nodeId)
```

Parameters

`nodeId` OpcNodeId

Returns

IOpcNode

Read(OpcContext, OpcNodeAccessTokenCollection)

C#

```
protected virtual void Read(OpcContext context, OpcNodeAccessTokenCollection tokens)
```

Parameters

`context` OpcContext

`tokens` OpcNodeAccessTokenCollection

ReadHistory(IOpcNode, Nullable<DateTime>, Nullable<DateTime>, OpcReadHistoryOptions)

C#

```
protected virtual IEnumerable<OpcHistoryValue> ReadHistory(IOpcNode node, DateTime?
startTime, DateTime? endTime, OpcReadHistoryOptions options)
```

Parameters

`node` IOpcNode

`startTime` Nullable<DateTime>

`endTime` Nullable<DateTime>

`options` OpcReadHistoryOptions

Returns

IEnumerable<OpcHistoryValue>

RemoveNode(IOpcNode)

C#

```
protected void RemoveNode(IOpcNode node)
```

Parameters

`node` IOpcNode

Exceptions

ArgumentNullException

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

RemoveNode(IOpcNode, OpcLocalReferenceCollection)

C#

```
protected void RemoveNode(IOpcNode node, OpcLocalReferenceCollection references)
```

Parameters

`node` IOpcNode

`references` OpcLocalReferenceCollection

Exceptions

ArgumentNullException

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

RemoveNode(OpcContext, IOpcNode)

C#

```
protected void RemoveNode(OpcContext context, IOpcNode node)
```

Parameters

`context` OpcContext

`node` IOpcNode

Exceptions

ArgumentNullException

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

RemoveNode(OpcContext, IOpcNode, OpcLocalReferenceCollection)

C#

```
protected virtual void RemoveNode(OpcContext context, IOpcNode node,
OpcLocalReferenceCollection references)
```

Parameters

`context` OpcContext

`node` IOpcNode

`references` OpcLocalReferenceCollection

Exceptions

ArgumentNullException

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

RemoveNotifierNode(IOpcNode)

C#

```
protected void RemoveNotifierNode(IOpcNode node)
```

Parameters

`node` IOpcNode

Exceptions

ArgumentNullException

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

ReplaceHistory(IOpcNode, OpcHistoryModificationInfo, OpcValueCollection)

C#

```
protected virtual OpcStatusCollection ReplaceHistory(IOpcNode node,
OpcHistoryModificationInfo modificationInfo, OpcValueCollection values)
```

Parameters

`node` IOpcNode

`modificationInfo` OpcHistoryModificationInfo

`values` OpcValueCollection

Returns

OpcStatusCollection

ReplaceNode(IOpcNode)

Replaces an existing IOpcNode instance with the specified `node` without invalidating existing monitored items nor previously defined notifiers.

C#

```
protected void ReplaceNode(IOpcNode node)
```

Parameters

`node` IOpcNode

The IOpcNode instance its Id is used to determine existing monitored items, notifiers and nodes to replace with the new one.

Exceptions

ArgumentNullException

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

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

Remarks

To replace an existing node the OpcNodeId of the `node` specified needs to match the Id of an existing IOpcNode instance.

This method replaces an existing node and all of its children recursively in the Nodes of this OpcNodeManager and in the list of notifiers as well in the list of monitored items / nodes.

The caller of this method have to ensure that the `node` specified is as compatible to the previous node as the clients of the server expect. This includes for example the attributes or other metrics of a dedicated type of node to which a client may belong.

ReportEvent(IOpcNode, OpcEventSeverity, OpcText)

Produces a global event using the event data specified.

C#

```
public void ReportEvent(IOpcNode source, OpcEventSeverity severity, OpcText message)
```

Parameters

`source` IOpcNode

The IOpcNode instance its Id is used for the SourceNodeId property and its SymbolicName is used for the SourceName property to define the origin of the event produced.

`severity` OpcEventSeverity

An indication of the urgency of the event.

`message` OpcText

A human-readable and localizable text description of the event.

Exceptions

ArgumentNullException

The `source` or `message` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

OpcException

The operation failed (see exception details for more information).

ReportEvent(IOpcNode, OpcEventSeverity, OpcText,

Object[])

Produces a global event using the event data specified.

C#

```
public void ReportEvent(IOpcNode source, OpcEventSeverity severity, OpcText message, params
object[] arguments)
```

Parameters

`source` IOpcNode

The IOpcNode instance its Id is used for the SourceNodeId property and its SymbolicName is used for the SourceName property to define the origin of the event produced.

`severity` OpcEventSeverity

An indication of the urgency of the event.

`message` OpcText

A human-readable and localizable text description of the event with format placeholders which where replaced with the `arguments` specified.

`arguments` Object[]

The values to place into the `message` using its format placeholders.

Exceptions

ArgumentNullException

The `source` or `message` is a null reference (Nothing in Visual Basic).

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

OpcException

The operation failed (see exception details for more information).

ReportEvent(OpcEventNode)

Produces a global event using the `eventNode` specified.

C#

```
public void ReportEvent(OpcEventNode eventNode)
```

Parameters

`eventNode` OpcEventNode

The OpcEventNode its event information is to be reported as a global event.

Exceptions

ArgumentNullException

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

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

OpcException

The operation failed (see exception details for more information).

ReportEvent(OpcEventSeverity, OpcText)

Produces a global event using the event data specified.

C#

```
public void ReportEvent(OpcEventSeverity severity, OpcText message)
```

Parameters

`severity` OpcEventSeverity

An indication of the urgency of the event.

`message` OpcText

A human-readable and localizable text description of the event.

Exceptions

ArgumentNullException

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

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

OpcException

The operation failed (see exception details for more information).

ReportEvent(OpcEventSeverity, OpcText, Object[])

Produces a global event using the event data specified.

C#

```
public void ReportEvent(OpcEventSeverity severity, OpcText message, params object[]
arguments)
```

Parameters

`severity` OpcEventSeverity

An indication of the urgency of the event.

`message` OpcText

A human-readable and localizable text description of the event with format placeholders which where replaced with the `arguments` specified.

`arguments` Object[]

The values to place into the `message` using its format placeholders.

Exceptions

ArgumentNullException

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

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

OpcException

The operation failed (see exception details for more information).

ReportEvent(OpcNodeId, String, OpcEventSeverity, OpcText)

Produces a global event using the event data specified.

C#

```
public void ReportEvent(OpcNodeId sourceNodeId, string sourceName, OpcEventSeverity
severity, OpcText message)
```

Parameters

`sourceNodeId` OpcNodeId

The OpcNodeId which is to be used for the SourceNodeId property to define the origin of the event produced.

`sourceName` String

The String which is to be used for the SourceName.

`severity` OpcEventSeverity

An indication of the urgency of the event.

`message` OpcText

A human-readable and localizable text description of the event.

Exceptions

ArgumentNullException

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

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

OpcException

The operation failed (see exception details for more information).

ReportEvent(OpcNodeId, String, OpcEventSeverity, OpcText, Object[])

Produces a global event using the event data specified.

C#

```
public void ReportEvent(OpcNodeId sourceNodeId, string sourceName, OpcEventSeverity
severity, OpcText message, params object[] arguments)
```

Parameters

`sourceNodeId` OpcNodeId

The OpcNodeId which is to be used for the SourceNodeId property to define the origin of the event produced.

`sourceName` String

The String which is to be used for the SourceName.

`severity` OpcEventSeverity

An indication of the urgency of the event.

`message` OpcText

A human-readable and localizable text description of the event with format placeholders which where replaced with the `arguments` specified.

`arguments` Object[]

The values to place into the `message` using its format placeholders.

Exceptions

ArgumentNullException

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

InvalidOperationException

The master node has not yet been created, therefore the operation is at the moment not available.

OpcException

The operation failed (see exception details for more information).

RetrieveNodeHistoryProvider(IOpcNode)

C#

```
protected virtual IOpcNodeHistoryProvider RetrieveNodeHistoryProvider(IOpcNode node)
```

Parameters

`node` IOpcNode

Returns

IOpcNodeHistoryProvider

UpdateHistory(IOpcNode, OpcHistoryModificationInfo, OpcValueCollection)

C#

```
protected virtual OpcStatusCollection UpdateHistory(IOpcNode node,
OpcHistoryModificationInfo modificationInfo, OpcValueCollection values)
```

Parameters

`node` IOpcNode

`modificationInfo` OpcHistoryModificationInfo

`values` OpcValueCollection

Returns

OpcStatusCollection

Write(OpcContext, OpcNodeAccessTokenCollection)

C#

```
protected virtual void Write(OpcContext context, OpcNodeAccessTokenCollection tokens)
```

Parameters

`context` OpcContext

`tokens` OpcNodeAccessTokenCollection

Table of Contents