

OpcEventNode Members

Namespace: Opc.UaFx

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

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

Constructors

OpcEventNode(IOpcNode, OpcName)

Initializes a new instance of the [OpcEventNode](#) class accessible by the **name** specified as a child node of the **parent** node given.

C#

```
public OpcEventNode(IOpcNode parent, OpcName name)
```

Parameters

parent [IOpcNode](#)

The [IOpcNode](#) used as the parent node or a null reference (Nothing in Visual Basic) in the case there is no parent node available.

name [OpcName](#)

The [OpcName](#) through that the new event node can be accessed.

OpcEventNode(IOpcNode, OpcName, OpcNodeId)

Initializes a new instance of the [OpcEventNode](#) class accessible by the **name** and **id** specified as a child node of the **parent** node given.

C#

```
public OpcEventNode(IOpcNode parent, OpcName name, OpcNodeId id)
```

Parameters

parent [IOpcNode](#)

The [IOpcNode](#) used as the parent node or a null reference (Nothing in Visual Basic) in the case there is no parent node available.

name [OpcName](#)

The [OpcName](#) through that the new event node can be accessed.

id [OpcNodeId](#)

The [OpcNodeId](#) through that the new event node can be identified and accessed.

OpcEventNode(OpcName)

Initializes a new instance of the [OpcEventNode](#) class accessible by the **name** specified.

C#

```
public OpcEventNode(OpcName name)
```

Parameters

name [OpcName](#)

The [OpcName](#) through that the new event node can be accessed.

OpcEventNode(OpcName, OpcNodeId)

Initializes a new instance of the [OpcEventNode](#) class accessible by the **name** and **id** specified.

C#

```
public OpcEventNode(OpcName name, OpcNodeId id)
```

Parameters

name [OpcName](#)

The [OpcName](#) through that the new event node can be accessed.

id [OpcNodeId](#)

The [OpcNodeId](#) through that the new event node can be identified and accessed.

Properties

DefaultTypeDefinitionId

Gets the default identifier which identifies the node that defines the underlying node type from that this [OpcInstanceNode](#) has been created.

C#

```
protected override OpcNodeId DefaultTypeDefinitionId { get; }
```

Property Value

[OpcNodeId](#)

The [OpcNodeId](#) of the type node from that this [OpcInstanceNode](#) has been created from. These type node defines the typical structure of an instance node of its type definition. If there exists no specific type definition node a null reference (Nothing in Visual Basic).

EventId

Gets or sets a value generated by the server to uniquely identify a particular event notification.

C#

```
public byte[] EventId { get; set; }
```

Property Value

Byte[]

The server is responsible to ensure that each event has its unique event identifier. It may do this, for example, by putting GUIDs into the ByteString. Clients can use the event identifier to assist in minimizing or eliminating gaps and overlaps that may occur during a redundancy failover. The event identifier shall always be returned as value and the server is not allowed to return a status code for the event identifier indicating an error.

EventIdNode

Gets the [OpcPropertyNode`1](#) of the [EventId](#) property.

C#

```
public OpcPropertyNode<byte[]> EventIdNode { get; }
```

Property Value

OpcPropertyNode<Byte>

An instance of the [OpcPropertyNode`1](#) class.

EventTypeId

Gets or sets the node identifier which describes the specific type of event.

C#

```
public OpcNodeId EventTypeId { get; set; }
```

Property Value

OpcNodeId

The event type shall always be returned as value and the server is not allowed to return a status code for the event type indicating an error.

EventTypeIdNode

Gets the [OpcNodeIdPropertyNode](#) of the [EventTypeId](#) property.

C#

```
public OpcNodeIdPropertyNode EventTypeIdNode { get; }
```

Property Value

[OpcNodeIdPropertyNode](#)

An instance of the [OpcNodeIdPropertyNode](#) class.

Message

Gets or sets a value which defines a human-readable and localizable text description of the event.

C#

```
public OpcText Message { get; set; }
```

Property Value

[OpcText](#)

The server may return any appropriate text to describe the event. A null reference (Nothing in Visual Basic) if it is not a valid value; if the server does not have a description, it shall return the string part of the [Name](#) of the node associated with the event.

MessageNode

Gets the [OpcTextPropertyNode](#) of the [Message](#) property.

C#

```
public OpcTextPropertyNode MessageNode { get; }
```

Property Value

[OpcTextPropertyNode](#)

An instance of the [OpcTextPropertyNode](#) class.

ReceiveTime

Gets or sets the time the OPC UA server received the event from the underlying device of another server.

C#

```
public DateTime ReceiveTime { get; set; }
```

Property Value

[DateTime](#)

The **ReceiveTime** is analogous to the **ServerTimestamp**, i.e. in the case where the OPC UA server gets an event from another OPC UA server, each server applies its own **ReceiveTime**. That implies that a client may get the same event, having the same **EventId**, from different servers having different values of the **ReceiveTime**. The **ReceiveTime** shall always be returned as value and the server is not allowed to return a status code for the **ReceiveTime** indicating an error.

ReceiveTimeNode

Gets the **OpcPropertyNode`1** of the **ReceiveTime** property.

C#

```
public OpcPropertyNode<DateTime> ReceiveTimeNode { get; }
```

Property Value

OpcPropertyNode<DateTime>

An instance of the **OpcPropertyNode`1** class.

Severity

Gets or sets an indication of the urgency of the event.

C#

```
public OpcEventSeverity Severity { get; set; }
```

Property Value

OpcEventSeverity

This value is commonly called “priority”. Values will range from 1 to 1 000, with 1 being the lowest severity and 1 000 being the highest. Typically, a severity of 1 would indicate an event which is informational in nature, while a value of 1 000 would indicate an event of catastrophic nature, which could potentially result in severe financial loss or loss of life.

Remarks

It is expected that very few server implementations will support 1 000 distinct severity levels. Therefore, server developers are responsible for distributing their severity levels across the 1 to 1 000 range in such a manner that clients can assume a linear distribution.

In many cases a strict linear mapping of underlying source severities to the OPC severity range is not appropriate. The server developer will instead intelligently map the underlying source severities to the 1 to 1 000 OPC severity range in some other fashion. In particular, it is recommended that server developers map events of high urgency into the OPC severity range of 667 to 1 000, events of medium urgency into the OPC severity range of 334 to 666 and events of low urgency into OPC severities of 1 to 333.

Some servers might not support any events which are catastrophic in nature, so they may choose to map all of their severities into a subset of the 1 to 1 000 range (for example, 1 to 666). Other servers might not support any events which are merely informational, so they may choose to map all of their severities into a different subset of the 1 to 1 000 range (for example, 334 to 1 000).

The purpose of this approach is to allow clients to use severity values from multiple servers from different vendors in a consistent manner.

SeverityNode

Gets the [OpcPropertyNode`1](#) of the [Severity](#) property.

C#

```
[CLSCompliant(false)]
public OpcPropertyNode<ushort> SeverityNode { get; }
```

Property Value

[OpcPropertyNode<UInt16>](#)

An instance of the [OpcPropertyNode`1](#) class.

SourceName

Gets or sets a description of the source of the event.

C#

```
public string SourceName { get; set; }
```

Property Value

[String](#)

A value which can be a string-part of the [DisplayName](#) of the event source using the default locale of the server, if the event is specific to a node, or some server-specific notation.

SourceNameNode

Gets the [OpcPropertyNode`1](#) of the [SourceName](#) property.

C#

```
public OpcPropertyNode<string> SourceNameNode { get; }
```

Property Value

[OpcPropertyNode<String>](#)

An instance of the [OpcPropertyNode`1](#) class.

SourceNodeId

Gets or sets the node identifier which identifies the node that the event originated from.

C#

```
public OpcNodeId SourceNodeId { get; set; }
```

Property Value

[OpcNodeId](#)

If the event is not specific to a node the value is a null reference (Nothing in Visual Basic); otherwise the [OpcNodeId](#) of the node that the event originated from. Some subtypes of this [OpcEventNode](#) may define additional rules for the [SourceNodeId](#).

SourceNodeIdNode

Gets the [OpcNodeIdPropertyNode](#) of the [SourceNodeId](#) property.

C#

```
public OpcNodeIdPropertyNode SourceNodeIdNode { get; }
```

Property Value

[OpcNodeIdPropertyNode](#)

An instance of the [OpcNodeIdPropertyNode](#) class.

Time

Gets or sets the time the event occurred.

C#

```
public DateTime Time { get; set; }
```

Property Value

[DateTime](#)

This value is set as close to the event generator as possible. It often comes from the underlying system or device. Once set, intermediate OPC UA servers shall not alter the value.

TimeNode

Gets the [OpcPropertyNode`1](#) of the [Time](#) property.

C#

```
public OpcPropertyNode<DateTime> TimeNode { get; }
```

Property Value

[OpcPropertyNode<DateTime>](#)

An instance of the [OpcPropertyNode`1](#) class.

Methods

CreateEvent(OpcContext)

Creates a new instance of the [OpcEvent](#) class which represents a snapshot of this [OpcEventNode](#).

C#

```
public OpcEvent CreateEvent(OpcContext context)
```

Parameters

context [OpcContext](#)

The [OpcContext](#) to use.

Returns

[OpcEvent](#)

A new immutable instance of the [OpcEvent](#) class representing a snapshot of the event node data at the time this method has been called.

CreateEventCore(OpcContext, OpcEventNodeSnapshot)

Creates a new instance of the [OpcEvent](#) class representing the specified **snapshot** of this [OpcEventNode](#).

C#

```
protected virtual OpcEvent CreateEventCore(OpcContext context, OpcEventNodeSnapshot snapshot)
```

Parameters

context [OpcContext](#)

The [OpcContext](#) used to create the **snapshot**.

snapshot [OpcEventNodeSnapshot](#)

The [OpcEventNodeSnapshot](#) taken from the current state of this [OpcEventNode](#) from that the new

[OpcEvent](#) is to be created.

Returns

[OpcEvent](#)

A new instance of the [OpcEvent](#) representing an immutable [snapshot](#) of this [OpcEventNode](#).

Remarks

In case there this method is not overridden the method will determine the type of [OpcEvent](#) to use using the [OpcEventTypeAttribute](#) of the [OpcEventNode](#) and will use this one to create a new [OpcEvent](#) using the [snapshot](#) specified.

InitializeDefaults()

Initializes the default values used by the [OpcEventNode](#).

C#

```
protected override void InitializeDefaults()
```

Remarks

This method is used to ensure the availability of appropriate node specific default values. For more information like when this method is to be overwritten see [InitializeDefaults](#).

ReportEvent(OpcContext)

Produces an event using the data represented by this [OpcEventNode](#).

C#

```
public void ReportEvent(OpcContext context)
```

Parameters

[context](#) [OpcContext](#)

The [OpcContext](#) to use.

Exceptions

[ArgumentNullException](#)

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

ReportEventFrom(OpcContext, IOpcNode)

Produces an event using the data represented by this [OpcEventNode](#) using the [source](#) as initiator of the event reported.

C#

```
public void ReportEventFrom(OpcContext context, IOpcNode source)
```

Parameters

[context](#) [OpcContext](#)

The [OpcContext](#) to use.

[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.

Exceptions

[ArgumentNullException](#)

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

Table of Contents

Constructors	1
OpcEventNode(IOpcNode, OpcName)	1
OpcEventNode(IOpcNode, OpcName, OpcNodeId)	1
OpcEventNode(OpcName)	2
OpcEventNode(OpcName, OpcNodeId)	2
Properties	2
DefaultTypeDefinitionId	2
EventId	3
EventIdNode	3
EventTypeId	3
EventTypeIdNode	3
Message	4
MessageNode	4
ReceiveTime	4
ReceiveTimeNode	5
Severity	5
SeverityNode	6
SourceName	6
SourceNameNode	6
SourceNodeId	7
SourceNodeIdNode	7
Time	7
TimeNode	7
Methods	8
CreateEvent(OpcContext)	8
CreateEventCore(OpcContext, OpcEventNodeSnapshot)	8
InitializeDefaults()	9
ReportEvent(OpcContext)	9
ReportEventFrom(OpcContext, IOpcNode)	10

