

IOpcWriteNodesService Members

Namespace: Opc.UaFx.Services

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

The [IOpcWriteNodesService](#) interface defines the following members.

Properties

MaxNodesPerWrite

Gets or sets the maximum number of nodes which shall be written using a single write request. This number indirectly defines the number of requests used to send a specific number of write requests to the server.

C#

```
uint? MaxNodesPerWrite { get; set; }
```

Property Value

[Nullable<UInt32>](#)

The absolute number of write commands which shall be send to the server per request or a null reference (Nothing in Visual Basic). In case there a specific number of nodes is used, a request with more commands as the [MaxNodesPerWrite](#) is partitioned into multiple requests and the responses of them are accumulated. The default value of this property is a null reference (Nothing in Visual Basic) and means that all commands defined are send to the server using the server defined number of nodes per write request.

TimestampSource

Gets or sets the [OpcTimestampSource](#) to use to prepare [OpcValue](#) timestamp values before sending a write request to a server.

C#

```
OpcTimestampSource? TimestampSource { get; set; }
```

Property Value

[Nullable<OpcTimestampSource>](#)

One of the members defined by the [OpcTimestampSource](#) enumeration; otherwise a null reference (Nothing in Visual Basic). In case there the value is null reference (Nothing in Visual Basic) there will be tried to automatically determine the timestamp sourcing to match the needs of the server. The default value of this property is a null reference (Nothing in Visual Basic).

UsedMaxNodesPerWrite

Gets the maximum number of nodes which are written using a single write request. This property will use either the [MaxNodesPerWrite](#) or the maximum number of nodes which can be written in one request as defined by the server. The lowest number of nodes is used to partition the commands to execute into multiple write requests.

C#

```
uint? UsedMaxNodesPerWrite { get; }
```

Property Value

[Nullable<UInt32>](#)

In case there has already been one write attempt (with more than one write command) the definite number of commands per write request; otherwise a null reference (Nothing in Visual Basic).

UsedTimestampSource

Gets the [OpcTimestampSource](#) currently used to prepare [OpcValue](#) timestamp values before sending a write request to a server. In case there [TimestampSource](#) is a null reference (Nothing in Visual Basic) this property will define the [OpcTimestampSource](#) automatically determined during the first write attempt which does write an [OpcValue](#) of the [Value](#) attribute.

C#

```
OpcTimestampSource? UsedTimestampSource { get; }
```

Property Value

[Nullable<OpcTimestampSource>](#)

In case there has already been one write attempt to a [Value](#) attribute the definite timestamp source; otherwise a null reference (Nothing in Visual Basic).

Methods

WriteNodes(IEnumerable<OpcWriteNode>)

Writes one or more values to one or more attributes of one or more nodes using the specified [commands](#).

C#

```
OpcStatusCollection WriteNodes(IEnumerable<OpcWriteNode> commands)
```

Parameters

[commands](#) [IEnumerable<OpcWriteNode>](#)

A sequence of [OpcWriteNode](#) instances to process.

Returns

[OpcStatusCollection](#)

An instance of the [OpcStatusCollection](#) its number and order of items matches the number and order of items in the [commands](#) sequence. There is for each command one entry which indicates the outcome of the operation for that command.

Exceptions

[ArgumentNullException](#)

The [commands](#) sequence or one of its items is a null reference (Nothing in Visual Basic).

[InvalidOperationException](#)

The write service is currently not supported, especially using the [commands](#) specified.

[OpcException](#)

The service execution failed upon different circumstances; for more details see exception details. The following issues can lead to that exception: [BadNothingToDo](#) and [BadTooManyOperations](#).

Remarks

Known results for the [commands](#) specified are:

Status	Description
GoodCompletesAsynchronously	The value was successfully written to an intermediate system but the Server does not know if the data source was updated properly.
BadNodeIdInvalid	For more details see BadNodeIdInvalid .
BadNodeIdUnknown	For more details see BadNodeIdUnknown .
BadAttributeInvalid	For more details see BadAttributeInvalid .
BadIndexRangeInvalid	For more details see BadIndexRangeInvalid .
BadIndexRangeNoData	For more details see BadIndexRangeNoData .
BadWriteNotSupported	The requested write operation is not supported. If a client attempts to write any value, quality, timestamp combination and the server does not support the requested combination (which could be a single quantity such as just timestamp), then the Server shall not perform any write on this node and shall return this status for that node.
BadNotWritable	For more details see BadNotWritable .
BadUserAccessDenied	For more details see BadUserAccessDenied .
BadOutOfRange	If a client attempts to write a value outside the valid range like a value not contained in the enumeration data type of a node, the server shall return this status for that node.
BadTypeMismatch	For more details see BadTypeMismatch .
BadDataEncodingUnsupported	For more details see BadDataEncodingUnsupported .
BadNoCommunication	For more details see BadNoCommunication .

WriteNodes(OpcWriteNode[])

Writes one or more values to one or more attributes of one or more nodes using the specified **commands**.

C#

```
OpcStatusCollection WriteNodes(params OpcWriteNode[] commands)
```

Parameters

commands [OpcWriteNode\[\]](#)

An array of [OpcWriteNode](#) instances to process.

Returns

[OpcStatusCollection](#)

An instance of the [OpcStatusCollection](#) its number and order of items matches the number and order of items in the **commands**. There is for each command one entry which indicates the outcome of the operation for that command.

Exceptions

[ArgumentNullException](#)

The **commands** array or one of its items is a null reference (Nothing in Visual Basic).

[InvalidOperationException](#)

The write service is currently not supported, especially using the **commands** specified.

[OpcException](#)

The service execution failed upon different circumstances; for more details see exception details. The following issues can lead to that exception: [BadNothingToDo](#) and [BadTooManyOperations](#).

Remarks

Known results for the **commands** specified are:

Status	Description
GoodCompletesAsynchronously	The value was successfully written to an intermediate system but the Server does not know if the data source was updated properly.
BadNodeIdInvalid	For more details see BadNodeIdInvalid .
BadNodeIdUnknown	For more details see BadNodeIdUnknown .
BadAttributeInvalid	For more details see BadAttributeInvalid .
BadIndexRangeInvalid	For more details see BadIndexRangeInvalid .
BadIndexRangeNoData	For more details see BadIndexRangeNoData .

Status	Description
BadWriteNotSupported	The requested write operation is not supported. If a client attempts to write any value, quality, timestamp combination and the server does not support the requested combination (which could be a single quantity such as just timestamp), then the Server shall not perform any write on this node and shall return this status for that node.
BadNotWritable	For more details see BadNotWritable .
BadUserAccessDenied	For more details see BadUserAccessDenied .
BadOutOfRange	If a client attempts to write a value outside the valid range like a value not contained in the enumeration data type of a node, the server shall return this status for that node.
BadTypeMismatch	For more details see BadTypeMismatch .
BadDataEncodingUnsupported	For more details see BadDataEncodingUnsupported .
BadNoCommunication	For more details see BadNoCommunication .

Table of Contents

Properties	1
MaxNodesPerWrite	1
TimestampSource	1
UsedMaxNodesPerWrite	2
UsedTimestampSource	2
Methods	2
WriteNodes(IEnumerable<OpcWriteNode>)	2
WriteNodes(OpcWriteNode[])	4