OpcStatus Members

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

Constructors

OpcStatus()

Initializes a new instance of the OpcStatus class which represents a 'neutral good' status.

C#

```
public OpcStatus()
```

OpcStatus(OpcStatusCode)

Initializes a new instance of the OpcStatus class which represents the information associated with the `code` specified.

C#

```
[CLSCompliant(false)]
public OpcStatus(OpcStatusCode code)
```

Parameters

`code` OpcStatusCode

The OpcStatusCode the information associated with the value is to be represented.

Events

Changed

Occurs when the OpcStatus changes.

C#

```
public event EventHandler Changed
```

Properties

AggregateBits

Gets a value indicating where aggregated data comes from.

C#

```
public OpcStatusAggregateBits AggregateBits { get; }
```

Property Value

OpcStatusAggregateBits

A combination of the members defined by the OpcStatusAggregateBits enumeration.

Code

Gets the underlying status code its information is represented by the OpcStatus.

C#

```
[CLSCompliant(false)]
public OpcStatusCode Code { get; }
```

Property Value

OpcStatusCode

One of the members defined by the OpcStatusCode enumeration.

CodeBits

Gets the bits (16:31) of the Code used to provide status data without any extra information.

C#

```
[CLSCompliant(false)]
public uint CodeBits { get; }
```

Property Value

UInt32

The 16 bits used to provide status data without any extra information.

Description

Gets a value which describes the status in detail.

C#

```
public string Description { get; }
```

Property Value

String

A String which describes the OpcStatusCode represented in detail.

FlagBits

Gets the bits (0:15) of the Code used to provide additional information about the status.

C#

```
[CLSCompliant(false)]
public uint FlagBits { get; }
```

Property Value

UInt32

The 16 bits used to provide additional information about the status.

HasSemanticsChanged

Gets a value indicating that the semantics of the associated data value have changed. Clients should not process the data value until they re-read the metadata associated with the variable. Servers should set this property to true if the metadata has changed in way that could cause application errors if the Client does not re-read the metadata. This is for example a change to the engineering units.

C#

```
public bool HasSemanticsChanged { get; }
```

Property Value

Boolean

The value true if the semantic information of a value have changed and a Client needs to re-read the metadata before processing the data value; otherwise the value false.

Remarks

Other specifications may define additional conditions. A Server may define other conditions that cause this property to be equals true. This property is only used as part of data change notifications or historical read operations. In other contexts this property shall be equals false.

HasStructureChanged

Gets a value indicating that the structure of the associated data value has changed since the last notification. Clients should not process the data value unless they re-read the metadata. Servers shall set this property to true if the data type encoding used for a variable or the enumeration information (via

EnumStrings) of the data type of a variable changes.

C#

```
public bool HasStructureChanged { get; }
```

Property Value

Boolean

The value true if the Server wants to warn Clients, that parse complex data values, that their parsing routines could fail because the serialized form of the data value changed; otherwise the value false.

Remarks

This property is only used as part of data change notifications or historical read operations. In other contexts this property shall be equals false.

HasValueInfo

Gets a value indicating that the status information is associated with a data value returned from the Server.

C#

```
public bool HasValueInfo { get; }
```

Property Value

Boolean

The value true if the status information belongs to a data value returned from the Server; otherwise the value false.

IsBad

Gets a value indicating whether the status Code expresses a bad outcome of the operation associated with this OpcStatus. In the most cases a Client has performed an invalid or (at the moment) unsupported operation. Another reason for a bad status could be that the Server may fail to complete the requested operation upon internal issues with the underlying data source. For more details about specific bad status information see the “bad” members of the OpcStatusCode enumeration.

C#

```
public bool IsBad { get; }
```

Property Value

Boolean

The value true if the Code expresses a failure; otherwise the value false.

IsGood

Gets a value indicating whether the status Code expresses a good outcome of the operation associated with this OpcStatus. The Client can be sure about that the Server successfully completed the operation requested and its result is totally reliable. Specific “good” status codes may inform the Client about the circumstances under which the operation completed. For more details about specific good status information see the “good” members of the OpcStatusCode enumeration.

C#

```
public bool IsGood { get; }
```

Property Value

Boolean

The value true if the Code expresses a good outcome; otherwise the value false.

IsOverflow

Gets a value indicating that not every detected change has been returned since the Servers queue buffer of a monitored item reached its limit and had to purge out data.

C#

```
public bool IsOverflow { get; }
```

Property Value

Boolean

The value true if the queue buffer of a monitored item reached its limit and had to purge out data.

Remarks

This property is only equals true if the monitored item queue size is greater than one.

IsUncertain

Gets a value indicating whether the status Code expresses an uncertain outcome of the operation associated with this OpcStatus. An operation with an uncertain outcome expresses in general that the Server can not guarantee a total reliable data value nor an entire complete and successful processing of an operation. The reasons to uncertains varying, for more details about specific uncertain status information see the “uncertain” members of the OpcStatusCode enumeration.

C#

```
public bool IsUncertain { get; }
```

Property Value

Boolean

The value true if the Code expresses an uncertain and there not reliable outcome; otherwise the value false.

LimitBits

Gets a value indicating the constraints a value falls within.

C#

```
public OpcStatusLimitBits LimitBits { get; }
```

Property Value

OpcStatusLimitBits

A combination of the members defined by the OpcStatusLimitBits enumeration.

SubCode

Gets the code used to represent different conditions.

C#

```
[CLSCompliant(false)]
public uint SubCode { get; }
```

Property Value

UInt32

A value representing the condition under that the status was produced.

Methods

CompareTo(Object)

Compares the current OpcStatus with the `other`.

C#

```
public int CompareTo(object other)
```

Parameters

`other` Object

The OpcStatus to compare with this OpcStatus.

Returns

Int32

A 32-bit signed integer that indicates the relative order of the objects being compared (CompareTo(Object)).

CompareTo(OpcStatus)

Compares the current OpcStatus with another OpcStatus.

C#

```
public int CompareTo(OpcStatus other)
```

Parameters

`other` OpcStatus

The OpcStatus to compare with this OpcStatus.

Returns

Int32

A 32-bit signed integer that indicates the relative order of the objects being compared (CompareTo()).

Equals(Object)

Determines whether the specified `other` is equal to this OpcStatus.

C#

```
public override bool Equals(object other)
```

Parameters

`other` Object

The OpcStatus to compare to the current OpcStatus.

Returns

Boolean

The value true if the specified OpcStatus is equal to the current OpcStatus; otherwise the value false.

Equals(OpcStatus)

Determines whether the specified `other` is equal to this OpcStatus.

C#

```
public bool Equals(OpcStatus other)
```

Parameters

`other` OpcStatus

The OpcStatus to compare to the current OpcStatus.

Returns

Boolean

The value true if the specified OpcStatus is equal to the current OpcStatus; otherwise the value false.

GetHashCode()

Retrieves a hash code for this OpcStatus.

C#

```
public override int GetHashCode()
```

Returns

Int32

An Int32 that contains the hash code for the OpcStatus.

OnChanged(EventArgs)

Raises the Changed event of the OpcStatus.

C#

```
protected virtual void OnChanged(EventArgs e)
```

Parameters

`e` EventArgs

The event data.

ToString()

Returns a string that represents the current OpcStatus.

C#

```
public override string ToString()
```

Returns

String

A string that represents the current OpcStatus including the Code and Description of the status represented.

Update(OpcStatus)

Updates the OpcStatus using the status data of the `other`OpcStatus.

C#

```
public void Update(OpcStatus other)
```

Parameters

`other` OpcStatus

The other OpcStatus its status information is to be adopted.

Remarks

Updating the OpcStatus results into Changed events.

Update(OpcStatusCode)

Updates the Code represented by the OpcStatus using the `code` specified.

C#

```
[CLSCompliant(false)]
public void Update(OpcStatusCode code)
```

Parameters

`code` OpcStatusCode

The OpcStatusCode to represent.

Remarks

Updating the OpcStatus results into Changed events.

Operators

Equality(OpcStatus, OpcStatus)

Returns a value indicating whether two instance of OpcStatus are equal.

C#

```
public static bool operator ==(OpcStatus left, OpcStatus right)
```

GreaterThan(OpcStatus, OpcStatus)

Determines whether the first specified OpcStatus object is greater than the second specified OpcStatus object.

C#

```
public static bool operator>(OpcStatus left, OpcStatus right)
```

GreaterThanOrEqual(OpcStatus, OpcStatus)

Determines whether the first specified OpcStatus object is greater than or equal to the second specified OpcStatus object.

C#

```
public static bool operator >=(OpcStatus left, OpcStatus right)
```

Implicit(OpcStatusCode to OpcStatus)

Converts a OpcStatusCode to an OpcStatus object.

C#

```
[CLSCompliant(false)]
public static implicit operator OpcStatus(OpcStatusCode value)
```

Implicit(StatusCode to OpcStatus)

Converts a StatusCode to an OpcStatus object.

C#

```
[CLSCompliant(false)]
public static implicit operator OpcStatus(StatusCode value)
```

Inequality(OpcStatus, OpcStatus)

Returns a value indicating whether two instances of OpcStatus are not equal.

C#

```
public static bool operator !=(OpcStatus left, OpcStatus right)
```

LessThan(OpcStatus, OpcStatus)

Determines whether the first specified OpcStatus object is less than the second specified OpcStatus object.

C#

```
public static bool operator <(OpcStatus left, OpcStatus right)
```

Exceptions

ArgumentNullException

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

LessThanOrEqual(OpcStatus, OpcStatus)

Determines whether the first specified OpcStatus object is less than or equal to the second OpcStatus object.

C#

```
public static bool operator <=(OpcStatus left, OpcStatus right)
```

Exceptions

ArgumentNullException

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

Table of Contents

Constructors 1
OpcStatus() 1
OpcStatus(OpcStatusCode) 1
Events 1
Changed 1
Properties 1
AggregateBits 2
Code 2
CodeBits 2
Description 2
FlagBits 3
HasSemanticsChanged 3
HasStructureChanged 3
HasValueInfo 4
IsBad 4
IsGood 5
IsOverflow 5
IsUncertain 5
LimitBits 6
SubCode 6
Methods 6
CompareTo(Object) 6
CompareTo(OpcStatus) 7
Equals(Object) 7
Equals(OpcStatus) 8
GetHashCode() 8
OnChanged(EventArgs) 8
ToString() 8
Update(OpcStatus) 9
Update(OpcStatusCode) 9
Operators 10
Equality(OpcStatus, OpcStatus) 10
GreaterThan(OpcStatus, OpcStatus) 10
GreaterThanOrEqual(OpcStatus, OpcStatus) 10
Implicit(OpcStatusCode to OpcStatus) 10
Implicit(StatusCode to OpcStatus) 10
Inequality(OpcStatus, OpcStatus) 11
LessThan(OpcStatus, OpcStatus) 11
LessThanOrEqual(OpcStatus, OpcStatus) 11