OpcStatus Class

Namespace: Opc.UaFx
Assemblies: Opc.UaFx.Advanced.dll

Provides information about the outcome of an operation.

C#

```
public class OpcStatus : IEquatable<OpcStatus>, IComparable, IComparable<OpcStatus>
```

Inheritance Object › OpcStatus

Implements IEquatable<OpcStatus>, IComparable, IComparable<OpcStatus>

Constructors

Name	Description
OpcStatus	Initializes a new instance of the OpcStatus class which represents a 'neutral good' status.
OpcStatus(OpcStatusCode)	Initializes a new instance of the OpcStatus class which represents the information associated with the `code` specified.

Events

Name	Description
Changed	Occurs when the OpcStatus changes.

Properties

Name	Description
AggregateBits	Gets a value indicating where aggregated data comes from.
Code	Gets the underlying status code its information is represented by the OpcStatus.
CodeBits	Gets the bits (16:31) of the Code used to provide status data without any extra information.
Description	Gets a value which describes the status in detail.
FlagBits	Gets the bits (0:15) of the Code 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.
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.

Name	Description
HasValueInfo	Gets a value indicating that the status information is associated with a data value returned from the Server.
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.
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.
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.
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.
LimitBits	Gets a value indicating the constraints a value falls within.
SubCode	Gets the code used to represent different conditions.

Methods

Name	Description
CompareTo(Object)	Compares the current OpcStatus with the `other`.
CompareTo(OpcStatus)	Compares the current OpcStatus with another OpcStatus.
Equals(Object)	Determines whether the specified `other` is equal to this OpcStatus.
Equals(OpcStatus)	Determines whether the specified `other` is equal to this OpcStatus.
GetHashCode	Retrieves a hash code for this OpcStatus.
OnChanged(EventArgs)	Raises the Changed event of the OpcStatus.
ToString	Returns a string that represents the current OpcStatus.
Update(OpcStatus)	Updates the OpcStatus using the status data of the `other`OpcStatus.
Update(OpcStatusCode)	Updates the Code represented by the OpcStatus using the `code` specified.

Operators

Name	Description
op_Equality(OpcStatus, OpcStatus)	Returns a value indicating whether two instance of OpcStatus are equal.
op_GreaterThan(OpcStatus, OpcStatus)	Determines whether the first specified OpcStatus object is greater than the second specified OpcStatus object.
op_GreaterThanOrEqual(OpcStatus, OpcStatus)	Determines whether the first specified OpcStatus object is greater than or equal to the second specified OpcStatus object.

Name	Description
op_Implicit(UaFx.OpcStatus)	Converts a OpcStatusCode to an OpcStatus object.
op_Implicit(StatusCode)~Opc.UaFx.OpcStatus)	Converts a StatusCode to an OpcStatus object.
op_Inequality(OpcStatus, OpcStatus)	Returns a value indicating whether two instances of OpcStatus are not equal.
op_LessThan(OpcStatus, OpcStatus)	Determines whether the first specified OpcStatus object is less than the second specified OpcStatus object.
op_LessThanOrEqual(OpcStatus, OpcStatus)	Determines whether the first specified OpcStatus object is less than or equal to the second OpcStatus object.

Table of Contents

Constructors 1
Events 1
Properties 1
Methods 2
Operators 2