

OpcValue Members

Namespace: Opc.UaFx

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

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

Constructors

OpcValue(Object)

Initializes a new instance of the [OpcValue](#) class using the [value](#).

C#

```
public OpcValue(object value)
```

Parameters

[value](#) [Object](#)

The value to represent.

OpcValue(Object, DateTime)

Initializes a new instance of the [OpcValue](#) class using the [value](#) and the [sourceTimestamp](#) of the [value](#).

C#

```
public OpcValue(object value, DateTime sourceTimestamp)
```

Parameters

[value](#) [Object](#)

The value to represent.

[sourceTimestamp](#) [DateTime](#)

The timestamp at which the source produced the [value](#).

OpcValue(Object, DateTime, DateTime)

Initializes a new instance of the [OpcValue](#) class using the [value](#), [sourceTimestamp](#) and [serverTimestamp](#) of the [value](#).

C#

```
public OpcValue(object value, DateTime sourceTimestamp, DateTime serverTimestamp)
```

Parameters

value Object

The value to represent.

sourceTimestamp DateTime

The timestamp at which the source produced the **value**.

serverTimestamp DateTime

The timestamp at which the server recorded the **value**.

OpcValue(Object, DateTime, DateTime, OpcStatus)

Initializes a new instance of the **OpcValue** class using the **value**, **sourceTimestamp**, **serverTimestamp** and **status** of the **value**.

C#

```
public OpcValue(object value, DateTime sourceTimestamp, DateTime serverTimestamp, OpcStatus status)
```

Parameters

value Object

The value to represent.

sourceTimestamp DateTime

The timestamp at which the source produced the **value**.

serverTimestamp DateTime

The timestamp at which the server recorded the **value**.

status OpcStatus

The status information associated with the **value**.

OpcValue(Object, DateTime, DateTime, OpcStatusCode)

Initializes a new instance of the **OpcValue** class using the **value**, **sourceTimestamp**, **serverTimestamp** and **statusCode** of the **value**.

C#

```
[CLSCompliant(false)]
public OpcValue(object value, DateTime sourceTimestamp, DateTime serverTimestamp, OpcStatusCode statusCode)
```

Parameters

value Object

The value to represent.

sourceTimestamp DateTime

The timestamp at which the source produced the value.

serverTimestamp DateTime

The timestamp at which the server recorded the value.

statusCode OpcStatusCode

One of the OpcStatusCode members which defines the status information associated with the value.

OpcValue(Object, DateTime, OpcStatus)

Initializes a new instance of the OpcValue class using the value, sourceTimestamp and status of the value.

C#

```
public OpcValue(object value, DateTime sourceTimestamp, OpcStatus status)
```

Parameters

value Object

The value to represent.

sourceTimestamp DateTime

The timestamp at which the source produced the value.

status OpcStatus

The status information associated with the value.

OpcValue(Object, DateTime, OpcStatusCode)

Initializes a new instance of the OpcValue class using the value, sourceTimestamp and statusCode of the value.

C#

```
[CLSCompliant(false)]
public OpcValue(object value, DateTime sourceTimestamp, OpcStatusCode statusCode)
```

Parameters

value Object

The value to represent.

sourceTimestamp DateTime

The timestamp at which the source produced the value.

[statusCode](#) [OpcStatusCode](#)

One of the [OpcStatusCode](#) members which defines the status information associated with the [value](#).

Properties

DataType

Gets the type of value represented.

C#

```
public OpcDataType DataType { get; }
```

Property Value

[OpcDataType](#)

One of the members defined by the [OpcDataType](#). In case there [Value](#) is a null reference (Nothing in Visual Basic) [Null](#).

DataTypeId

Gets the node identifier of the [DataType](#) of the value represented.

C#

```
public virtual OpcNodeId DataTypeId { get; }
```

Property Value

[OpcNodeId](#)

An instance of the [OpcNodeId](#) representing the node identifier of the [DataType](#).

Rank

Gets the rank of the value represented.

C#

```
public int Rank { get; }
```

Property Value

[Int32](#)

The rank of the value or -2 in case there is no type information available for the value represented.

ServerPicoseconds

Gets or sets the difference between a high resolution timestamp with a resolution of 10 picoseconds and the [ServerTimestamp](#) which only has a 100 ns resolution.

C#

```
public ushort ServerPicoseconds { get; set; }
```

Property Value

[UInt16](#)

The number of 10 picosecond intervals for the [ServerTimestamp](#).

Remarks

The value should be less than 10 000.

ServerTimestamp

Gets or sets the timestamp at which the server recorded the value.

C#

```
public DateTime? ServerTimestamp { get; set; }
```

Property Value

[Nullable<DateTime>](#)

The timestamp of the server for the value or a null reference (Nothing in Visual Basic) in case there is no timestamp provided by the server.

SourcePicoseconds

Gets or sets the difference between a high resolution timestamp with a resolution of 10 picoseconds and the [SourceTimestamp](#) which only has a 100 ns resolution.

C#

```
public ushort SourcePicoseconds { get; set; }
```

Property Value

[UInt16](#)

The number of 10 picosecond intervals for the [SourceTimestamp](#).

Remarks

The value should be less than 10 000.

SourceTimestamp

Gets or sets the timestamp of the source from that the value originates.

C#

```
public DateTime? SourceTimestamp { get; set; }
```

Property Value

[Nullable<DateTime>](#)

The timestamp of the source for the value or a null reference (Nothing in Visual Basic) in case there is no timestamp provided by the source of the value.

Status

Gets the status information which defines the servers ability to access/provide the value.

C#

```
public OpcStatus Status { get; }
```

Property Value

[OpcStatus](#)

The status information associated with the value.

Remarks

If the status information indicates an error then the value is to be ignored and the server shall set it to a null reference (Nothing in Visual Basic).

Value

Gets or sets the value represented.

C#

```
public object Value { get; set; }
```

Property Value

[Object](#)

The value associated with the metadata represented.

Methods

As<T>(T)

Retrieves the underlying **Value** as the type specified by **T**.

C#

```
public T As<T>(T defaultValue = null)
```

Parameters

defaultValue **T**

The value to provide in the case there the **Value** is a null reference (Nothing in Visual Basic).

Returns

T

The **Value** typed as **T** or **defaultValue** in case there **Value** is a null reference (Nothing in Visual Basic) or could not converted to the **T** specified.

AsValue<T>()

Retrieves a new **OpcValue`1** instance its **Value** is of the type specified by **T**.

C#

```
public OpcValue<T> AsValue<T>()
```

Returns

OpcValue<T>

A new instance of the **OpcValue`1** class its **Value** is typed as **T** and its further value information has been adopted from this **OpcValue** or a null reference (Nothing in Visual Basic) if the **Value** could not converted to the **T** specified.

CompareTo(Object)

Compares the current **OpcValue** with the **other**.

C#

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

Parameters

other Object

The [OpcValue](#) to compare with this [OpcValue](#).

Returns

[Int32](#)

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

CompareTo(OpcValue)

Compares the current [OpcValue](#) with another [OpcValue](#).

C#

```
public virtual int CompareTo(OpcValue other)
```

Parameters

other [OpcValue](#)

The [OpcValue](#) to compare with this [OpcValue](#).

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 [OpcValue](#).

C#

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

Parameters

other Object

The [OpcValue](#) to compare to the current [OpcValue](#).

Returns

[Boolean](#)

The value true if the specified [OpcValue](#) is equal to the current [OpcValue](#); otherwise the value false.

Equals(OpcValue)

Determines whether the specified **other** is equal to this **OpcValue**.

C#

```
public virtual bool Equals(OpcValue other)
```

Parameters

other **OpcValue**

The **OpcValue** to compare to the current **OpcValue**.

Returns

Boolean

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

GetHashCode()

Retrieves a hash code for this **OpcValue**.

C#

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

Returns

Int32

An **Int32** that contains the hash code for the **OpcValue**.

ToString()

Converts the value of this instance to its equivalent string representation.

C#

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

Returns

String

The string representation of the value of this instance.

ToString(String, IFormatProvider)

Converts the value of this instance to its equivalent string representation using the specified format and

culture-specific format information.

C#

```
public virtual string ToString(string format, IFormatProvider formatProvider)
```

Parameters

format [String](#)

A standard or custom format string.

formatProvider [IFormatProvider](#)

An object that supplies culture-specific formatting information.

Returns

[String](#)

The string representation of the value of this instance as specified by format and provider.

Exceptions

[FormatException](#)

The **format** is invalid or not supported.

Operators

Equality(OpcValue, OpcValue)

Returns a value indicating whether two instance of [OpcValue](#) are equal.

C#

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

Explicit(OpcValue to DataValue)

Converts a [OpcValue](#) to an [DataValue](#) object.

C#

```
[CLSCompliant(false)]  
public static explicit operator DataValue(OpcValue value)
```

GreaterThan(OpcValue, OpcValue)

Determines whether the first specified [OpcValue](#) object is greater than the second specified [OpcValue](#) object.

C#

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

GreaterThanOrEqual(OpcValue, OpcValue)

Determines whether the first specified [OpcValue](#) object is greater than or equal to the second specified [OpcValue](#) object.

C#

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

Implicit(Boolean to OpcValue)

Converts a [Boolean](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(bool value)
```

Implicit(Byte to OpcValue)

Converts a [Byte](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(byte value)
```

Implicit(Char to OpcValue)

Converts a [Char](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(char value)
```

Implicit(DataValue to OpcValue)

Converts a [DataValue](#) to an [OpcValue](#) object.

C#

```
[CLSCompliant(false)]  
public static implicit operator OpcValue(DataValue value)
```

Implicit(Decimal to OpcValue)

Converts a [Decimal](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(decimal value)
```

Implicit(Double to OpcValue)

Converts a [Double](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(double value)
```

Implicit(Int16 to OpcValue)

Converts a [UInt16](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(short value)
```

Implicit(Int32 to OpcValue)

Converts a [Int32](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(int value)
```

Implicit(Int64 to OpcValue)

Converts a [UInt64](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(long value)
```

Implicit(Nullable<Boolean> to OpcValue)

Converts a null-able [Boolean](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(bool? value)
```

Implicit(Nullable<Byte> to OpcValue)

Converts a null-able [Byte](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(byte? value)
```

Implicit(Nullable<Char> to OpcValue)

Converts a null-able [Char](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(char? value)
```

Implicit(Nullable<Decimal> to OpcValue)

Converts a null-able [Decimal](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(decimal? value)
```

Implicit(Nullable<Double> to OpcValue)

Converts a null-able [Double](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(double? value)
```

Implicit(Nullable<Int16> to OpcValue)

Converts a null-able [UInt16](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(short? value)
```

Implicit(Nullable<Int32> to OpcValue)

Converts a null-able [Int32](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(int? value)
```

Implicit(Nullable<Int64> to OpcValue)

Converts a null-able [UInt64](#) to an [OpcValue](#) object.

C#

```
[CLSCompliant(false)]  
public static implicit operator OpcValue(long? value)
```

Implicit(Nullable<SByte> to OpcValue)

Converts a null-able [SByte](#) to an [OpcValue](#) object.

C#

```
[CLSCompliant(false)]  
public static implicit operator OpcValue(sbyte? value)
```

Implicit(Nullable<Single> to OpcValue)

Converts a null-able [Single](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(float? value)
```

Implicit(Nullable<UInt16> to OpcValue)

Converts a null-able [UInt16](#) to an [OpcValue](#) object.

C#

```
[CLSCompliant(false)]  
public static implicit operator OpcValue(ushort? value)
```

Implicit(Nullable<UInt32> to OpcValue)

Converts a null-able [UInt32](#) to an [OpcValue](#) object.

C#

```
[CLSCompliant(false)]  
public static implicit operator OpcValue(uint? value)
```

Implicit(Nullable<UInt64> to OpcValue)

Converts a null-able [UInt64](#) to an [OpcValue](#) object.

C#

```
[CLSCompliant(false)]  
public static implicit operator OpcValue(ulong? value)
```

Implicit(SByte to OpcValue)

Converts a [SByte](#) to an [OpcValue](#) object.

C#

```
[CLSCompliant(false)]  
public static implicit operator OpcValue(sbyte value)
```

Implicit(Single to OpcValue)

Converts a [Single](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(float value)
```

Implicit(String to OpcValue)

Converts a [String](#) to an [OpcValue](#) object.

C#

```
public static implicit operator OpcValue(string value)
```

Implicit(UInt16 to OpcValue)

Converts a [UInt16](#) to an [OpcValue](#) object.

C#

```
[CLSCompliant(false)]  
public static implicit operator OpcValue(ushort value)
```

Implicit(UInt32 to OpcValue)

Converts a [UInt32](#) to an [OpcValue](#) object.

C#

```
[CLSCompliant(false)]  
public static implicit operator OpcValue(uint value)
```

Implicit(UInt64 to OpcValue)

Converts a [UInt64](#) to an [OpcValue](#) object.

C#

```
[CLSCompliant(false)]
public static implicit operator OpcValue(ulong value)
```

Implicit(Variant to OpcValue)

Converts a [Variant](#) to an [OpcValue](#) object.

C#

```
[CLSCompliant(false)]
public static implicit operator OpcValue(Variant value)
```

Inequality(OpcValue, OpcValue)

Returns a value indicating whether two instances of [OpcValue](#) are not equal.

C#

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

LessThan(OpcValue, OpcValue)

Determines whether the first specified [OpcValue](#) object is less than the second specified [OpcValue](#) object.

C#

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

Exceptions

[ArgumentNullException](#)

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

LessThanOrEqual(OpcValue, OpcValue)

Determines whether the first specified [OpcValue](#) object is less than or equal to the second [OpcValue](#) object.

C#

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

Exceptions

[ArgumentNullException](#)

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

Table of Contents

Constructors	1
OpcValue(Object)	1
OpcValue(Object, DateTime)	1
OpcValue(Object, DateTime, DateTime)	1
OpcValue(Object, DateTime, DateTime, OpcStatus)	2
OpcValue(Object, DateTime, DateTime, OpcStatusCode)	2
OpcValue(Object, DateTime, OpcStatus)	3
OpcValue(Object, DateTime, OpcStatusCode)	3
Properties	4
DataType	4
DataTypeeld	4
Rank	4
ServerPicoseconds	5
ServerTimestamp	5
SourcePicoseconds	5
SourceTimestamp	6
Status	6
Value	6
Methods	7
As<T>(T)	7
AsValue<T>()	7
CompareTo(Object)	7
CompareTo(OpcValue)	8
Equals(Object)	8
Equals(OpcValue)	9
GetHashCode()	9
ToString()	9
ToString(String, IFormatProvider)	9
Operators	10
Equality(OpcValue, OpcValue)	10
Explicit(OpcValue to DataValue)	10
GreaterThan(OpcValue, OpcValue)	11
GreaterThanOrEqual(OpcValue, OpcValue)	11
Implicit(Boolean to OpcValue)	11
Implicit(Byte to OpcValue)	11
Implicit(Char to OpcValue)	11
Implicit(DataValue to OpcValue)	11
Implicit(Decimal to OpcValue)	12
Implicit(Double to OpcValue)	12
Implicit(Int16 to OpcValue)	12
Implicit(Int32 to OpcValue)	12
Implicit(Int64 to OpcValue)	12
Implicit(Nullable<Boolean> to OpcValue)	12
Implicit(Nullable<Byte> to OpcValue)	13
Implicit(Nullable<Char> to OpcValue)	13
Implicit(Nullable<Decimal> to OpcValue)	13
Implicit(Nullable<Double> to OpcValue)	13
Implicit(Nullable<Int16> to OpcValue)	13
Implicit(Nullable<Int32> to OpcValue)	13
Implicit(Nullable<Int64> to OpcValue)	14
Implicit(Nullable<SByte> to OpcValue)	14

Implicit(Nullable<Single> to OpcValue)	14
Implicit(Nullable<UInt16> to OpcValue)	14
Implicit(Nullable<UInt32> to OpcValue)	14
Implicit(Nullable<UInt64> to OpcValue)	14
Implicit(SByte to OpcValue)	15
Implicit(Single to OpcValue)	15
Implicit(String to OpcValue)	15
Implicit(UInt16 to OpcValue)	15
Implicit(UInt32 to OpcValue)	15
Implicit(UInt64 to OpcValue)	15
Implicit(Variant to OpcValue)	16
Inequality(OpcValue, OpcValue)	16
LessThan(OpcValue, OpcValue)	16
LessThanOrEqual(OpcValue, OpcValue)	16