

# OpcValue<T> Members

**Namespace:** Opc.UaFx

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

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

## Constructors

### OpcValue(T)

Initializes a new instance of the [OpcValue`1](#) class using the [value](#).

**C#**

```
public OpcValue(T value)
```

#### Parameters

[value](#) T

The value to represent.

### OpcValue(T, DateTime)

Initializes a new instance of the [OpcValue`1](#) class using the [value](#) and the [sourceTimestamp](#) of the [value](#).

**C#**

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

#### Parameters

[value](#) T

The value to represent.

[sourceTimestamp](#) DateTime

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

### OpcValue(T, DateTime, DateTime)

Initializes a new instance of the [OpcValue`1](#) class using the [value](#), [sourceTimestamp](#) and [serverTimestamp](#) of the [value](#).

**C#**

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

## Parameters

value T

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(T, DateTime, DateTime, OpcStatus)

Initializes a new instance of the `OpcValue`1` class using the value, sourceTimestamp, serverTimestamp and status of the value.

### C#

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

## Parameters

value T

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(T, DateTime, DateTime, OpcStatusCode)

Initializes a new instance of the `OpcValue`1` class using the value, sourceTimestamp, serverTimestamp and statusCode of the value.

### C#

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

## Parameters

value T

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(T, DateTime, OpcStatus)

Initializes a new instance of the OpcValue'1 class using the value, sourceTimestamp and status of the value.

C#

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

### Parameters

value T

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(T, DateTime, OpcStatusCode)

Initializes a new instance of the OpcValue'1 class using the value, sourceTimestamp and statusCode of the value.

C#

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

### Parameters

value T

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

## DataTypeId

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

**C#**

```
public override sealed OpcNodeId DataTypeId { get; }
```

### Property Value

[OpcNodeId](#)

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

## Value

Gets or sets the [T](#) value represented.

**C#**

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

### Property Value

[T](#)

The value of the type [T](#) associated with the metadata represented.

## ValueAsObject

Gets or sets the value represented.

**C#**

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

### Property Value

[Object](#)

The value associated with the metadata represented.

# Methods

## CompareTo(Object)

Compares the current [OpcValue`1](#) with the [other](#).

**C#**

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

### Parameters

[other](#) [Object](#)

The [OpcValue`1](#) to compare with this [OpcValue`1](#).

### 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`1](#) with another [OpcValue`1](#).

**C#**

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

### Parameters

[other](#) [OpcValue](#)

The [OpcValue`1](#) to compare with this [OpcValue`1](#).

### Returns

[Int32](#)

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

## CompareTo(OpcValue<T>)

Compares the current [OpcValue`1](#) with another [OpcValue`1](#).

**C#**

```
public virtual int CompareTo(OpcValue<T> other)
```

## Parameters

other OpcValue`1

The OpcValue`1 to compare with this OpcValue`1.

## 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`1.

C#

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

## Parameters

other Object

The OpcValue`1 to compare to the current OpcValue`1.

## Returns

Boolean

The value true if the specified OpcValue`1 is equal to the current OpcValue; otherwise the value false.

# Equals(OpcValue)

Determines whether the specified other is equal to this OpcValue`1.

C#

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

## Parameters

other OpcValue

The OpcValue`1 to compare to the current OpcValue`1.

## Returns

Boolean

The value true if the specified OpcValue`1 is equal to the current OpcValue; otherwise the value false.

# Equals(OpcValue<T>)

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

**C#**

```
public virtual bool Equals(OpcValue<T> other)
```

## Parameters

**other** **OpcValue`1**

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`1**.

**C#**

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

## Returns

**Int32**

An **Int32** that contains the hash code for the **OpcValue`1**.

# Operators

## Equality(OpcValue<T>, OpcValue<T>)

Returns a value indicating whether two instance of **OpcValue`1** are equal.

**C#**

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

## Explicit(OpcValue<T> to DataValue)

Converts a **OpcValue`1** to an **DataValue** object.

**C#**

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

## GreaterThan(OpcValue<T>, OpcValue<T>)

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

**C#**

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

## GreaterThanOrEqual(OpcValue<T>, OpcValue<T>)

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

**C#**

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

## Implicit(DataValue to OpcValue<T>)

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

**C#**

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

## Implicit(Variant to OpcValue<T>)

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

**C#**

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

## Inequality(OpcValue<T>, OpcValue<T>)

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

**C#**

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



## LessThan(OpcValue<T>, OpcValue<T>)

Determines whether the first specified [OpcValue'1](#) object is less than the second specified [OpcValue'1](#) object.

### C#

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

### Exceptions

[ArgumentNullException](#)

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

## LessThanOrEqual(OpcValue<T>, OpcValue<T>)

Determines whether the first specified [OpcValue'1](#) object is less than or equal to the second [OpcValue'1](#) object.

### C#

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

### Exceptions

[ArgumentNullException](#)

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



# Table of Contents

<b>Constructors</b>	1
OpcValue(T)	1
OpcValue(T, DateTime)	1
OpcValue(T, DateTime, DateTime)	1
OpcValue(T, DateTime, DateTime, OpcStatus)	2
OpcValue(T, DateTime, DateTime, OpcStatusCode)	2
OpcValue(T, DateTime, OpcStatus)	3
OpcValue(T, DateTime, OpcStatusCode)	3
<b>Properties</b>	4
DataTypeId	4
Value	4
ValueAsObject	4
<b>Methods</b>	5
CompareTo(Object)	5
CompareTo(OpcValue)	5
CompareTo(OpcValue<T>)	5
Equals(Object)	6
Equals(OpcValue)	6
Equals(OpcValue<T>)	7
GetHashCode()	7
<b>Operators</b>	7
Equality(OpcValue<T>, OpcValue<T>)	7
Explicit(OpcValue<T> to DataValue)	7
GreaterThan(OpcValue<T>, OpcValue<T>)	8
GreaterThanOrEqual(OpcValue<T>, OpcValue<T>)	8
Implicit(DataValue to OpcValue<T>)	8
Implicit(Variant to OpcValue<T>)	8
Inequality(OpcValue<T>, OpcValue<T>)	8
LessThan(OpcValue<T>, OpcValue<T>)	9
LessThanOrEqual(OpcValue<T>, OpcValue<T>)	9