

PlcValue<T> Members

Namespace: IPS7Lnk.Advanced

Assemblies: IPS7LnkNet.Advanced.dll, IPS7LnkNet.Advanced.dll

The `PlcValue<T>` type exposes the following members.

Constructors

PlcValue(PlcType)

Initializes a new instance of the `PlcValue'1` class using the specified `type`.

C#

```
protected PlcValue(PlcType type)
```

Parameters

`type` `PlcType`

The type of value represented.

Exceptions

`ArgumentNullException`

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

PlcValue(PlcType, PlcName)

Initializes a new instance of the `PlcValue'1` class using the specified `type` and `name`.

C#

```
protected PlcValue(PlcType type, PlcName name)
```

Parameters

`type` `PlcType`

The type of value represented.

`name` `PlcName`

The name of the value represented.

Exceptions

`ArgumentNullException`

The `type` or `name` is a null reference (Nothing in Visual Basic).

PlcValue(PlcType, PlcName, T)

Initializes a new instance of the `PlcValue`1` class using the specified `type`, `name` and `value`.

C#

```
protected PlcValue(PlcType type, PlcName name, T value)
```

Parameters

`type` `PlcType`

The type of value represented.

`name` `PlcName`

The name of the value represented.

`value` `T`

The initial value of the new `PlcValue`1`.

Exceptions

`ArgumentNullException`

The `type` or `name` is a null reference (Nothing in Visual Basic).

PlcValue(PlcType, T)

Initializes a new instance of the `PlcValue`1` class using the specified `type` and `value`.

C#

```
protected PlcValue(PlcType type, T value)
```

Parameters

`type` `PlcType`

The type of value represented.

`value` `T`

The initial value of the new `PlcValue`1`.

Exceptions

`ArgumentNullException`

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

Events

Changed

Occurs when the value of the [Value](#) property has changed.

C#

```
public event ValueChangedEventHandler<T> Changed
```

Properties

Description

Gets or sets more meaningful information about the value and its usage than only using the [Name](#) property.

C#

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

Property Value

[String](#)

A [String](#) value containing additional information about the value and its usage.

IsInitializing

Gets a value indicating whether the instance is in initializing mode.

C#

```
protected bool IsInitializing { get; }
```

Property Value

[Boolean](#)

The value true, if the instance is in initializing mode; otherwise the value false.

Remarks

In case there the value is true, the data provided by [Value](#) represents the data before the initialization sequence has been entered. This means that any changes to the [Value](#) property will only take place after the initialization sequence has been exited.

Also any value validations are performed after the initialization sequence has been exited. This also includes the [Changed](#) event.

Name

Gets the name of the value represented.

C#

```
public PlcName Name { get; }
```

Property Value

[PlcName](#)

An instance of the [PlcName](#) class containing the name of the value represented.

Status

Gets the status information provided by the software driver related to the value.

C#

```
public PlcStatus Status { get; }
```

Property Value

[PlcStatus](#)

The latest status information provided by the software driver related to the value.

Tag

Gets or sets the object that contains additional user data about the value.

C#

```
public virtual object Tag { get; set; }
```

Property Value

[Object](#)

An [Object](#) that contains additional user data about the value. The default is null (Nothing in Visual Basic).

Type

Gets the type of value represented by the PLC value.

C#

```
public PlcType Type { get; }
```

Property Value

[PlcType](#)

The [PlcType](#) of the value represented by the PLC value.

Value

Gets or sets the value assigned to the PLC value.

C#

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

Property Value

[T](#)

The [T](#) value assigned to the PLC value.

Exceptions

[ArgumentException](#)

The [value](#) is invalid. It either does not fall within an expected value range or its type is not supported by the [PlcValue`1](#).

[ArgumentNullException](#)

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

Methods

BeginInitCore()

When implemented in a derived class; signals the [PlcValue`1](#) that initialization is started.

C#

```
protected virtual void BeginInitCore()
```

EndInitCore()

When implemented in a derived class; signals the [PlcValue`1](#) that initialization is completed.

C#

```
protected virtual void EndInitCore()
```

GetInstanceValue()

Retrieves the value assigned to the PLC value.

C#

```
protected virtual T GetInstanceValue()
```

Returns

T

The value assigned to the PLC value.

GetValue(IPlcDevice)

Retrieves the current value of the PLC value from the **device** specified.

C#

```
public T GetValue(IPlcDevice device)
```

Parameters

device [IPlcDevice](#)

The [IPlcDevice](#) from that the data is to be retrieved.

Returns

T

The current value of the PLC value.

Exceptions

[ArgumentNullException](#)

The **device** is a null reference (Nothing in Visual Basic).

GetValue(PlcDeviceConnection)

Retrieves the current value of the PLC value from a [IPlcDevice](#) using the **connection** specified.

C#

```
public T GetValue(PlcDeviceConnection connection)
```

Parameters

connection [PlcDeviceConnection](#)

The [PlcDeviceConnection](#) from that the data is to be retrieved.

Returns

[T](#)

The current value of the PLC value.

Exceptions

[ArgumentNullException](#)

The **connection** is a null reference (Nothing in Visual Basic).

[InvalidOperationException](#)

The **connection** is in [Faulted](#) state and cannot longer be opened.

[ObjectDisposedException](#)

The **connection** has been disposed of.

GetValueCore(PlcDeviceConnection)

Retrieves the current value of the PLC value from a [IPlcDevice](#) using the **connection** specified.

C#

```
protected virtual T GetValueCore(PlcDeviceConnection connection)
```

Parameters

connection [PlcDeviceConnection](#)

The [PlcDeviceConnection](#) from that the data is to be retrieved.

Returns

[T](#)

The current value of the PLC value.

Exceptions

[ArgumentNullException](#)

The **connection** is a null reference (Nothing in Visual Basic).

[InvalidOperationException](#)

The `connection` is in `Faulted` state and cannot longer be opened.

`ObjectDisposedException`

The `connection` has been disposed of.

IsValidType(Object)

Determines whether a specified value is acceptable for this `PlcValue<T>`.

C#

```
public virtual bool IsValidType(object value)
```

Parameters

`value` `Object`

The value to check.

Returns

`Boolean`

The value true, if the specified `value` is the `Type` or an acceptable derived type; otherwise the value false.

IsValidValue(Object)

Determines whether the provided value is accepted for the type of PLC value through basic type checking and also potentially if it is within the allowed range of value for that type.

C#

```
public virtual bool IsValidValue(object value)
```

Parameters

`value` `Object`

The value to check.

Returns

`Boolean`

The value true, if the specified `value` is acceptable and is of the correct type or a derived type; otherwise the value false.

OnChanged(ValueChangedEventArgs)

Raises the `Changed` event.

C#

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

Parameters

e ValueChangedEventArgs

The ValueChangedEventArgs that contains the event data.

OnChanged(ValueChangedEventArgs<T>)

Raises the Changed event.

C#

```
protected virtual void OnChanged(ValueChangedEventArgs<T> e)
```

Parameters

e ValueChangedEventArgs<T>

The ValueChangedEventArgs¹ that contains the event data.

RaiseChanged(T, T)

Raises the Changed and Changed events.

C#

```
protected virtual void RaiseChanged(T oldValue, T newValue)
```

Parameters

oldValue T

The value before the Value property gets changed.

newValue T

The value after the Value property gets changed.

Relocate(Int32)

Relocates the PlcValue¹ using the specified offset. The original PlcValue¹ remains unmodified.

C#

```
public IPlcValue Relocate(int operandNumber)
```

Parameters

operandNumber Int32

The operand number offset used to adjust the address of the [Type](#).

Returns

IPlcValue

A new instance of the [PlcValue`1](#) configured with the same metadata as this instance but relocated using the specified offset.

Exceptions

ArgumentOutOfRangeException

The offset specified by [operandNumber](#) result into a new value that would be out of the bounds defined by [MinOperandNumber](#) or [MaxOperandNumber](#).

InvalidOperationException

It is not possible to relocate relative types.

Relocate(Int32, Int32)

Relocates the [PlcValue`1](#) using the specified offset. The original [PlcValue`1](#) remains unmodified.

C#

```
public IPlcValue Relocate(int operandNumber, int byteNumber)
```

Parameters

operandNumber Int32

The operand number offset used to adjust the address of the [Type](#).

byteNumber Int32

The byte number offset used to adjust the address of the [Type](#).

Returns

IPlcValue

A new instance of the [PlcValue`1](#) configured with the same metadata as this instance but relocated using the specified offset.

Exceptions

ArgumentOutOfRangeException

One of the offsets specified by [operandNumber](#) or [byteNumber](#) result into a new value that would be out of

the bounds defined by [MinOperandNumber](#), [MaxOperandNumber](#), [MinByteNumber](#) and [MaxByteNumber](#).

[InvalidOperationException](#)

It is not possible to relocate relative types.

Relocate(Int32, Int32, Int32)

Relocates the [PlcValue`1](#) using the specified offset. The original [PlcValue`1](#) remains unmodified.

C#

```
public IPlcValue Relocate(int operandNumber, int byteNumber, int bitNumber)
```

Parameters

[operandNumber](#) [Int32](#)

The operand number offset used to adjust the address of the [Type](#).

[byteNumber](#) [Int32](#)

The byte number offset used to adjust the address of the [Type](#).

[bitNumber](#) [Int32](#)

The bit number offset used to adjust the address of the [Type](#).

Returns

[IPlcValue](#)

A new instance of the [PlcValue`1](#) configured with the same metadata as this instance but relocated using the specified offset.

Exceptions

[ArgumentOutOfRangeException](#)

One of the offsets specified by [operandNumber](#), [byteNumber](#) or [bitNumber](#) result into a new value that would be out of the bounds defined by [MinOperandNumber](#), [MaxOperandNumber](#), [MinByteNumber](#), [MaxByteNumber](#), [MinBitNumber](#) or [MaxBitNumber](#).

[InvalidOperationException](#)

It is not possible to relocate relative types.

Relocate(PlcAddress)

Relocates the [PlcValue`1](#) using the specified [address](#). The original [PlcValue`1](#) remains unmodified.

C#

```
public IPlcValue Relocate(PlcAddress address)
```

Parameters

address PlcAddress

The **PlcAddress** to that the value is to be relocated.

Returns

IPlcValue

A new instance of the **PlcValue** configured with the same metadata as this instance but relocated using the specified **address**.

Exceptions

ArgumentNullException

The **address** is a null reference (Nothing in Visual Basic).

ArgumentException

The **address** does not refer to the same **RawType** as this **PlcType**.

InvalidOperationException

It is not possible to relocate absolute object types without relative type information.

Remarks

The **RawType** of the **address** specified needs to be the same as defined by the address of the **Type** of the PLC value.

RelocateCore(PlcAddress)

When implemented in a derived class, relocates the value to the **address** specified.

C#

```
protected abstract IPlcValue RelocateCore(PlcAddress address)
```

Parameters

address PlcAddress

The **PlcAddress** to that the value is to be relocated.

Returns

IPlcValue

A new instance of the same type as this PLC value instance relocated to the **address** specified.

SetInstanceValue(T)

Applies the value to the PLC value.

C#

```
protected virtual void SetInstanceValue(T value)
```

Parameters

value T

The new value for the PLC value.

SetValue(IPlcDevice)

Stores the **Value** in the **device** specified.

C#

```
public void SetValue(IPlcDevice device)
```

Parameters

device IPlcDevice

The **IPlcDevice** in that the data is to be stored.

Exceptions

ArgumentNullException

The **device** is a null reference (Nothing in Visual Basic).

SetValue(IPlcDevice, T)

Stores the **value** in the **device** specified.

C#

```
public void SetValue(IPlcDevice device, T value)
```

Parameters

device IPlcDevice

The **IPlcDevice** in that the data is to be stored.

value T

The value to be stored.

Exceptions

ArgumentException

The **value** is invalid. It either does not fall within an expected value range or its type is not supported by the **PlcValue`1**.

ArgumentNullException

The **device** is a null reference (Nothing in Visual Basic).

SetValue(PlcDeviceConnection)

Stores the **Value** in the **IPlcDevice** assigned to the **connection** specified.

C#

```
public void SetValue(PlcDeviceConnection connection)
```

Parameters

connection **PlcDeviceConnection**

The **PlcDeviceConnection** to use to store the **Value**.

Exceptions

ArgumentNullException

The **connection** is a null reference (Nothing in Visual Basic).

InvalidOperationException

The **connection** is in **Faulted** state and cannot longer be opened.

ObjectDisposedException

The **connection** has been disposed of.

SetValue(PlcDeviceConnection, T)

Stores the **value** in the **IPlcDevice** assigned to the **connection** specified.

C#

```
public void SetValue(PlcDeviceConnection connection, T value)
```

Parameters

connection PlcDeviceConnection

The `PlcDeviceConnection` to use to store the `value`.

`value T`

The value to be stored.

Exceptions

ArgumentException

The `value` is invalid. It either does not fall within an expected value range or its type is not supported by the `PlcValue`1`.

ArgumentNullException

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

InvalidOperationException

The `connection` is in `Faulted` state and cannot longer be opened.

ObjectDisposedException

The `connection` has been disposed of.

SetValueCore(PlcDeviceConnection, T)

Stores the `value` in the `IPlcDevice` assigned to the `connection` specified.

C#

```
protected virtual void SetValueCore(PlcDeviceConnection connection, T value)
```

Parameters

connection PlcDeviceConnection

The `PlcDeviceConnection` to use to store the `value`.

`value T`

The value to be stored.

Exceptions

ArgumentException

The `value` is invalid. It either does not fall within an expected value range or its type is not supported by the `PlcValue`1`.

ArgumentNullException

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

InvalidOperationException

The **connection** is in **Faulted** state and cannot longer be opened.

ObjectDisposedException

The **connection** has been disposed of.

ToString()

Converts the value to its string representation.

C#

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

Returns

String

A string that contains the value.

ValidateValue(Object)

Validates the specified **value** whether it can be assigned to this **PlcValue`1**.

C#

```
public void ValidateValue(object value)
```

Parameters

value Object

The value to validate.

Exceptions

ArgumentException

The **value** is invalid. It either does not fall within an expected value range or its type is not supported by the **PlcValue`1**.

Operators

Explicit(PlcValue<T> to T)

Converts a **PlcValue`1** to an object of the type **T**.

C#


```
public static explicit operator T(PlcValue<T> value)
```


Table of Contents

Constructors	1
PlcValue(PlcType)	1
PlcValue(PlcType, PlcName)	1
PlcValue(PlcType, PlcName, T)	2
PlcValue(PlcType, T)	2
Events	3
Changed	3
Properties	3
Description	3
IsInitializing	3
Name	4
Status	4
Tag	4
Type	4
Value	5
Methods	5
BeginInitCore()	5
EndInitCore()	5
GetInstanceValue()	6
GetValue(IPlcDevice)	6
GetValue(PlcDeviceConnection)	6
GetValueCore(PlcDeviceConnection)	7
IsValidType(Object)	8
IsValidValue(Object)	8
OnChanged(ValueChangedEventArgs)	8
OnChanged(ValueChangedEventArgs<T>)	9
RaiseChanged(T, T)	9
Relocate(Int32)	9
Relocate(Int32, Int32)	10
Relocate(Int32, Int32, Int32)	11
Relocate(PlcAddress)	11
RelocateCore(PlcAddress)	12
SetInstanceValue(T)	13
SetValue(IPlcDevice)	13
SetValue(IPlcDevice, T)	13
SetValue(PlcDeviceConnection)	14
SetValue(PlcDeviceConnection, T)	14
SetValueCore(PlcDeviceConnection, T)	15
ToString()	16
ValidateValue(Object)	16
Operators	16
Explicit(PlcValue<T> to T)	16