

PlcOperand Members

Namespace: IPS7Lnk.Advanced

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

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

Constructors

PlcOperand(PlcOperandType)

Initializes a new instance of the [PlcOperand](#) class using the `type` specified.

C#

```
protected PlcOperand(PlcOperandType type)
```

Parameters

`type` [PlcOperandType](#)

The [PlcOperandType](#) of the operand to represent.

Exceptions

[ArgumentException](#)

The [DataBlock](#) can not be used to create a [PlcOperand](#), because it requires a number.

PlcOperand(PlcOperandType, Int32)

Initializes a new instance of the [PlcOperand](#) class using the `type` and `number` specified.

C#

```
protected PlcOperand(PlcOperandType type, int number)
```

Parameters

`type` [PlcOperandType](#)

The [PlcOperandType](#) of the operand to represent.

`number` [Int32](#)

The number of the block to represent.

Exceptions

[ArgumentException](#)

The `type` specified can not be used to create a `PlcOperand`, because it does not require a number.

ArgumentOutOfRangeException

The `number` is out of the bounds defined by `MinNumber` or `MaxNumber`.

PlcOperand(SerializationInfo, StreamingContext)

Initializes a new instance of the `PlcOperand` class with serialized data.

C#

```
protected PlcOperand(SerializationInfo info, StreamingContext context)
```

Parameters

`info` `Serialization.SerializationInfo`

The `Serialization.SerializationInfo` that holds the serialized object data about the exception being thrown.

`context` `Serialization.StreamingContext`

The `Serialization.StreamingContext` that contains contextual information about the source or destination.

Exceptions

`ArgumentNullException`

The `info` parameter is null reference (Nothing in Visual Basic).

`Serialization.SerializationException`

The class name is null reference (Nothing in Visual Basic) or `HResult` is zero (0).

Fields

MaxNumber

Specifies the maximum value that can be assigned to the `Number` property.

C#

```
public const int MaxNumber = 65535
```

Field Value

`Int32`

MinNumber

Specifies the minimum value that can be assigned to the `Number` property.

C#

```
public const int MinNumber =
```

Field Value

Int32

Properties

Counter

Gets the operand used to access the counter memory block of the controller.

C#

```
public static PlcCounterOperand Counter { get; }
```

Property Value

PlcCounterOperand

An instance of the [PlcCounterOperand](#) class.

Flag

Gets the operand used to access the flag memory block of the controller.

C#

```
public static PlcFlagOperand Flag { get; }
```

Property Value

PlcFlagOperand

An instance of the [PlcFlagOperand](#) class.

Input

Gets the operand used to access the input memory block of the controller.

C#

```
public static PlcInputOperand Input { get; }
```

Property Value

PlcInputOperand

An instance of the [PlcInputOperand](#) class.

Number

Gets the number associated with the operand.

C#

```
public int Number { get; }
```

Property Value

[Int32](#)

The value -1 if the [Type](#) is not equals to [DataBlock](#); otherwise a number within the range defined by [MinNumber](#) and [MaxNumber](#).

OriginalString

Gets the original string from that the operand was created.

C#

```
public string OriginalString { get; }
```

Property Value

[String](#)

The original [String](#) from that the operand was created or [Empty](#) in case there the operand was created using one of its constructors.

Output

Gets the operand used to access the output memory block of the controller.

C#

```
public static PlcOutputOperand Output { get; }
```

Property Value

[PlcOutputOperand](#)

An instance of the [PlcOutputOperand](#) class.

PeripheryInput

Gets the operand used to access the peripheryInput memory block of the controller.

C#

```
public static PlcPeripheryInputOperand PeripheryInput { get; }
```

Property Value

[PlcPeripheryInputOperand](#)

An instance of the [PlcPeripheryInputOperand](#) class.

PeripheryOutput

Gets the operand used to access the peripheryOutput memory block of the controller.

C#

```
public static PlcPeripheryOutputOperand PeripheryOutput { get; }
```

Property Value

[PlcPeripheryOutputOperand](#)

An instance of the [PlcPeripheryOutputOperand](#) class.

Standard

Gets a value indicating which [PlcOperandStandard](#) was used to create the operand.

C#

```
public PlcOperandStandard Standard { get; }
```

Property Value

[PlcOperandStandard](#)

One of the members defined by the [PlcOperandStandard](#) enumeration.

Timer

Gets the operand used to access the timer memory block of the controller.

C#

```
public static PlcTimerOperand Timer { get; }
```

Property Value

[PlcTimerOperand](#)

An instance of the [PlcTimerOperand](#) class.

Type

Gets the type of operand represented.

C#

```
public PlcOperandType Type { get; }
```

Property Value

[PlcOperandType](#)

One of the members defined by the [PlcOperandType](#) enumeration.

Methods

All()

Retrieves all [PlcOperand](#) instances which can be typically addressed within a PLC including the custom data block operands used so far.

C#

```
public static IEnumerable<PlcOperand> All()
```

Returns

[IEnumerable<PlcOperand>](#)

An [IEnumerable`1](#) of [PlcOperand](#) instances representing the possible [PlcOperand](#) instances including the custom data block operands.

CompareTo(Object)

Compares the current [PlcOperand](#) with the [other](#).

C#

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

Parameters

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

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

Returns

[Int32](#)

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

CompareTo(PlcOperand)

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

C#

```
public int CompareTo(PlcOperand other)
```

Parameters

other [PlcOperand](#)

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

Returns

[Int32](#)

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

DataBlock(Int32)

Retrieves the according [PlcDataBlockOperand](#) using the specified **number** of the data block operand.

C#

```
public static PlcDataBlockOperand DataBlock(int number)
```

Parameters

number [Int32](#)

The number of the data block to retrieve.

Returns

[PlcDataBlockOperand](#)

The [PlcDataBlockOperand](#) of a specific data block.

Exceptions

[ArgumentOutOfRangeException](#)

The **number** is out of the bounds defined by [MinNumber](#) or [MaxNumber](#).

Equals(Object)

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

C#

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

Parameters

`other Object`

The `PlcOperand` to compare to the current `PlcOperand`.

Returns

`Boolean`

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

Equals(PlcOperand)

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

C#

```
public bool Equals(PlcOperand other)
```

Parameters

`other PlcOperand`

The `PlcOperand` to compare to the current `PlcOperand`.

Returns

`Boolean`

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

GetHashCode()

Retrieves a hash code for this `PlcOperand`.

C#

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

Returns

`Int32`

An `Int32` that contains the hash code for the `PlcOperand`.

GetObjectData(SerializationInfo, StreamingContext)

Sets the `Serialization.SerializationInfo` with information about the exception.

C#

```
[SecurityPermission(SecurityAction.LinkDemand, Flags =  
SecurityPermissionFlag.SerializationFormatter)]  
public override void GetObjectData(SerializationInfo info, StreamingContext context)
```

Parameters

`info` `Serialization.SerializationInfo`

The `Serialization.SerializationInfo` that holds the serialized object data about the exception being thrown.

`context` `Serialization.StreamingContext`

The `Serialization.StreamingContext` that contains contextual information about the source or destination.

Exceptions

`ArgumentNullException`

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

GetSupportedRawTypes(PlcOperandType)

Retrieves the `PlcRawType` members supported by the `PlcOperandType` specified by `type`.

C#

```
public static PlcRawType[] GetSupportedRawTypes(PlcOperandType type)
```

Parameters

`type` `PlcOperandType`

The `PlcOperandType` its supported `PlcRawType` members are to be determined.

Returns

`PlcRawType[]`

An array of `PlcRawType` members supported by the `type` specified.

Remarks

Supported raw types means that these `PlcRawType` members can be used to address the PLC memory of

such a [PlcOperandType](#) by using one of the supported [PlcRawType](#) members.

IsSupported(PlcOperandType, PlcRawType)

Determines whether the specified [rawType](#) can be used to address the PLC memory identified by the [PlcOperandType](#) specified by [type](#).

C#

```
public static bool IsSupported(PlcOperandType type, PlcRawType rawType)
```

Parameters

[type](#) [PlcOperandType](#)

The [PlcOperandType](#) its supported raw types are inspected to determine the accessibility of the PLC memory using the [PlcRawType](#) specified by [rawType](#).

[rawType](#) [PlcRawType](#)

The [PlcRawType](#) its compatibility to address the PLC memory defined by [type](#) is to be inspected.

Returns

[Boolean](#)

The value true if the PLC memory of [type](#) can be accessed using the [PlcRawType](#) specified by [rawType](#); otherwise the value false.

Of(PlcOperandType)

Retrieves the according [PlcOperand](#) which matches the [PlcOperandType](#) specified.

C#

```
public static PlcOperand Of(PlcOperandType type)
```

Parameters

[type](#) [PlcOperandType](#)

The [PlcOperandType](#) of the [PlcOperand](#) to retrieve.

Returns

[PlcOperand](#)

The [PlcOperand](#) which matches the [type](#).

Exceptions

ArgumentException

The `type` is either a [DataBlock](#) or is an unknown [PlcOperandType](#).

Parse(String)

Converts an operand string to a [PlcOperand](#) instance.

C#

```
public static PlcOperand Parse(string value)
```

Parameters

`value` [String](#)

A string that contains an operand.

Returns

[PlcOperand](#)

An instance of the [PlcOperand](#) class.

Exceptions

[FormatException](#)

The `value` is not a valid operand.

ToString()

Converts the operand to its string representation.

C#

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

Returns

[String](#)

A string that contains the operand.

ToString(PlcOperandStandard)

Converts the operand to its string representation using the specified `standard`.

C#

```
public string ToString(PlcOperandStandard standard)
```

Parameters

standard PlcOperandStandard

The [PlcOperandStandard](#) to use to identify the standard dependent characters in the string representation of the operand.

Returns

[String](#)

A string that contains the operand.

TryParse(String, out PlcOperand)

Determines whether a string is a valid operand.

C#

```
public static bool TryParse(string value, out PlcOperand operand)
```

Parameters

value String

The string to validate.

operand PlcOperand

The [PlcOperand](#) version of the string.

Returns

[Boolean](#)

The value true, if **value** is a valid operand; otherwise the value false.

UserDefinedType(Int32)

Retrieves the according [PlcUserDefinedTypeOperand](#) using the specified **number** of the user defined type operand.

C#

```
public static PlcUserDefinedTypeOperand UserDefinedType(int number)
```

Parameters

number Int32

The number of the user defined type to retrieve.

Returns

[PlcUserDefinedTypeOperand](#)

The [PlcUserDefinedTypeOperand](#) of a specific user defined type.

Exceptions

[ArgumentOutOfRangeException](#)

The `number` is out of the bounds defined by [MinNumber](#) or [MaxNumber](#).

Operators

Equality(PlcOperand, PlcOperand)

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

C#

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

GreaterThan(PlcOperand, PlcOperand)

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

C#

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

GreaterThanOrEqual(PlcOperand, PlcOperand)

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

C#

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

Implicit(String to PlcOperand)

Converts a string formatted as operand to an [PlcOperand](#) object.

C#

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

Exceptions

FormatException

The `value` is not a valid PLC operand.

Inequality(PlcOperand, PlcOperand)

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

C#

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

LessThan(PlcOperand, PlcOperand)

Determines whether the first specified `PlcOperand` object is less than the second specified `PlcOperand` object.

C#

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

Exceptions

ArgumentNullException

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

LessThanOrEqual(PlcOperand, PlcOperand)

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

C#

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

Exceptions

ArgumentNullException

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

Table of Contents

Constructors	1
PlcOperand(PlcOperandType)	1
PlcOperand(PlcOperandType, Int32)	1
PlcOperand(SerializationInfo, StreamingContext)	2
Fields	2
MaxNumber	2
MinNumber	2
Properties	3
Counter	3
Flag	3
Input	3
Number	4
OriginalString	4
Output	4
PeripheryInput	4
PeripheryOutput	5
Standard	5
Timer	5
Type	6
Methods	6
All()	6
CompareTo(Object)	6
CompareTo(PlcOperand)	7
DataBlock(Int32)	7
Equals(Object)	8
Equals(PlcOperand)	8
GetHashCode()	8
GetObjectData(SerializationInfo, StreamingContext)	9
GetSupportedRawTypes(PlcOperandType)	9
IsSupported(PlcOperandType, PlcRawType)	10
Of(PlcOperandType)	10
Parse(String)	11
ToString()	11
ToString(PlcOperandStandard)	11
TryParse(String, out PlcOperand)	12
UserDefinedType(Int32)	12
Operators	13
Equality(PlcOperand, PlcOperand)	13
GreaterThan(PlcOperand, PlcOperand)	13
GreaterThanOrEqual(PlcOperand, PlcOperand)	13
Implicit(String to PlcOperand)	13
Inequality(PlcOperand, PlcOperand)	14
LessThan(PlcOperand, PlcOperand)	14
LessThanOrEqual(PlcOperand, PlcOperand)	14

