

PlcAddress Members

Namespace: IPS7Lnk.Advanced

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

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

Constructors

PlcAddress(PlcOperand, PlcRawType, Int32)

Initializes a new instance of the [PlcAddress](#) class using the specified [operand](#), [rawType](#) and [byteNumber](#).

C#

```
public PlcAddress(PlcOperand operand, PlcRawType rawType, int byteNumber)
```

Parameters

[operand](#) [PlcOperand](#)

The [PlcOperand](#) to that the new [PlcAddress](#) refers.

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

The [PlcRawType](#) to that the new [PlcAddress](#) refers.

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

The number of the byte to that the new [PlcAddress](#) refers.

Exceptions

[ArgumentException](#)

The type of [operand](#) specified is not supported or specifying a [rawType](#) equals to [Bit](#) requires a bit number.

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

The [byteNumber](#) is out of the bounds defined by [MinByteNumber](#) or [MaxByteNumber](#).

PlcAddress(PlcOperand, PlcRawType, Int32, Int32)

Initializes a new instance of the [PlcAddress](#) class using the specified [operand](#), [rawType](#), [byteNumber](#) and [bitNumber](#).

C#

```
public PlcAddress(PlcOperand operand, PlcRawType rawType, int byteNumber, int bitNumber)
```

Parameters

operand PlcOperand

The PlcOperand to that the new PlcAddress refers.

rawType PlcRawType

The PlcRawType to that the new PlcAddress refers.

byteNumber Int32

The number of the byte to that the new PlcAddress refers.

bitNumber Int32

The number of the bit to that the new PlcAddress refers.

Exceptions

ArgumentException

The type of **operand** specified is not supported or does not support a bit number or specifying a **bitNumber** requires the **rawTypeBit**.

ArgumentNullException

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

ArgumentOutOfRangeException

The **byteNumber** or **bitNumber** is out of the bounds defined by **MinByteNumber**, **MaxByteNumber**, **MinBitNumber** or **MaxBitNumber**.

PlcAddress(PlcOperandType, Int32, PlcRawType, Int32)

Initializes a new instance of the PlcAddress class using the specified **operandType**, **operandNumber**, **rawType** and **byteNumber**.

C#

```
public PlcAddress(PlcOperandType operandType, int operandNumber, PlcRawType rawType, int byteNumber)
```

Parameters

operandType PlcOperandType

The PlcOperandType to that the new PlcAddress refers.

operandNumber Int32

The number of the operand to that the new PlcAddress refers.

rawType PlcRawType

The PlcRawType to that the new PlcAddress refers.

byteNumber **Int32**

The number of the byte to that the new **PlcAddress** refers.

Exceptions

ArgumentException

The **operandType** specified is not supported or the **operandType** specified does not support an operand number or specifying a **rawType** equals to **Bit** requires a bit number.

ArgumentOutOfRangeException

The **operandNumber** or **byteNumber** is out of the bounds defined by **MinOperandNumber**, **MaxOperandNumber**, **MinByteNumber** or **MaxByteNumber**.

PlcAddress(PlcOperandType, Int32, PlcRawType, Int32, Int32)

Initializes a new instance of the **PlcAddress** class using the specified **operandType**, **operandNumber**, **rawType**, **byteNumber** and **bitNumber**.

C#

```
public PlcAddress(PlcOperandType operandType, int operandNumber, PlcRawType rawType, int byteNumber, int bitNumber)
```

Parameters

operandType **PlcOperandType**

The **PlcOperandType** to that the new **PlcAddress** refers.

operandNumber **Int32**

The number of the operand to that the new **PlcAddress** refers.

rawType **PlcRawType**

The **PlcRawType** to that the new **PlcAddress** refers.

byteNumber **Int32**

The number of the byte to that the new **PlcAddress** refers.

bitNumber **Int32**

The number of the bit to that the new **PlcAddress** refers.

Exceptions

ArgumentException

The **operandType** specified is not supported the **operandType** specified does not support an operand number or specifying a **bitNumber** requires the **rawTypeBit**.

ArgumentOutOfRangeException

The **operandNumber**, **byteNumber** or **bitNumber** is out of the bounds defined by **MinOperandNumber**, **MaxOperandNumber**, **MinByteNumber**, **MaxByteNumber**, **MinBitNumber** or **MaxBitNumber**.

PlcAddress(PlcOperandType, PlcRawType, Int32)

Initializes a new instance of the **PlcAddress** class using the specified **operandType**, **rawType**, and **byteNumber**.

C#

```
public PlcAddress(PlcOperandType operandType, PlcRawType rawType, int byteNumber)
```

Parameters

operandType **PlcOperandType**

The **PlcOperandType** to that the new **PlcAddress** refers.

rawType **PlcRawType**

The **PlcRawType** to that the new **PlcAddress** refers.

byteNumber **Int32**

The number of the byte to that the new **PlcAddress** refers.

Exceptions

ArgumentException

The **operandType** specified requires an operand number or specifying a **rawType** equals to **Bit** requires a bit number.

ArgumentOutOfRangeException

The **byteNumber** is out of the bounds defined by **MinByteNumber** or **MaxByteNumber**.

PlcAddress(PlcOperandType, PlcRawType, Int32, Int32)

Initializes a new instance of the **PlcAddress** class using the specified **operandType**, **rawType**, **byteNumber** and **bitNumber**.

C#

```
public PlcAddress(PlcOperandType operandType, PlcRawType rawType, int byteNumber, int bitNumber)
```

Parameters

operandType **PlcOperandType**

The **PlcOperandType** to that the new **PlcAddress** refers.

rawType PlcRawType

The [PlcRawType](#) to that the new [PlcAddress](#) refers.

byteNumber Int32

The number of the byte to that the new [PlcAddress](#) refers.

bitNumber Int32

The number of the bit to that the new [PlcAddress](#) refers.

Exceptions

ArgumentException

The [operandType](#) specified requires an operand number or does not support a bit number or specifying a [bitNumber](#) requires the [rawTypeBit](#).

ArgumentOutOfRangeException

The [byteNumber](#) or [bitNumber](#) is out of the bounds defined by [MinByteNumber](#), [MaxByteNumber](#), [MinBitNumber](#) or [MaxBitNumber](#).

PlcAddress(SerializationInfo, StreamingContext)

Initializes a new instance of the [PlcAddress](#) class with serialized data.

C#

```
protected PlcAddress(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

MaxBitNumber

Specifies the maximum value that can be assigned to the [BitNumber](#) property.

C#

```
public const int MaxBitNumber = 7
```

Field Value

[Int32](#)

MaxByteNumber

Specifies the maximum value that can be assigned to the [ByteNumber](#) property.

C#

```
public const int MaxByteNumber = 65535
```

Field Value

[Int32](#)

MaxOperandNumber

Specifies the maximum value that can be assigned to the [OperandNumber](#) property.

C#

```
public const int MaxOperandNumber = 65535
```

Field Value

[Int32](#)

MinBitNumber

Specifies the minimum value that can be assigned to the [BitNumber](#) property.

C#

```
public const int MinBitNumber =
```

Field Value

[Int32](#)

MinByteNumber

Specifies the minimum value that can be assigned to the [ByteNumber](#) property.

C#

```
public const int MinByteNumber =
```

Field Value

[Int32](#)

MinOperandNumber

Specifies the minimum value that can be assigned to the [OperandNumber](#) property.

C#

```
public const int MinOperandNumber =
```

Field Value

[Int32](#)

Properties

BitNumber

Gets the bit number part of the address, which defines to which bit the address refers.

C#

```
public int BitNumber { get; }
```

Property Value

[Int32](#)

The number indicating to which specific bit the address refers or -1 if [PlcRawType](#) is not equals to [Bit](#).

ByteNumber

Gets the byte number part of the address, which defines to which byte the address refers.

C#

```
public int ByteNumber { get; }
```

Property Value

Int32

The number indicating to which specific byte the address refers.

Operand

Gets the operand part of the address, which defines to which memory block the address refers.

C#

```
public PlcOperand Operand { get; }
```

Property Value

PlcOperand

The operand indicating to which specific memory block the address refers.

OperandNumber

Gets the operand number part of the address, which defines to which operand the address refers.

C#

```
public int OperandNumber { get; }
```

Property Value

Int32

The number indicating to which specific operand the address refers or -1 if [PlcOperandType](#) is not equals [DataBlock](#).

OperandType

Gets the type of operand to which the address refers.

C#

```
public PlcOperandType OperandType { get; }
```

Property Value

PlcOperandType

One of the members defined by the [PlcOperandType](#) enumeration. Which specifies the type of operand to which the address refers.

OriginalString

Gets the original string from that the address was created.

C#

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

Property Value

[String](#)

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

RawType

Gets the raw type of to which the address refers.

C#

```
public PlcRawType RawType { get; }
```

Property Value

[PlcRawType](#)

One of the members defined by the [PlcRawType](#) enumeration. Which specifies the type to which the address refers.

Standard

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

C#

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

Property Value

[PlcOperandStandard](#)

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

Methods

CompareTo(Object)

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

C#

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

Parameters**other** Object

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

Returns[Int32](#)

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

CompareTo(PlcAddress)

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

C#

```
public int CompareTo(PlcAddress other)
```

Parameters**other** [PlcAddress](#)

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

Returns[Int32](#)

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

Distinct(IEnumerable<PlcAddress>)

Returns distinct addresses from the sequence.

C#

```
public static IEnumerable<PlcAddress> Distinct(IEnumerable<PlcAddress> addresses)
```

Parameters**addresses** [IEnumerable<PlcAddress>](#)

The sequence to remove duplicate addresses from.

Returns

[IEnumerable<PlcAddress>](#)

An [IEnumerable<T>](#) that contains distinct addresses from the source sequence.

Exceptions

[ArgumentNullException](#)

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

Equals(Object)

Determines whether the specified [other](#) is equal to this [PlcAddress](#).

C#

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

Parameters

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

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

Returns

[Boolean](#)

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

Equals(PlcAddress)

Determines whether the specified [other](#) is equal to this [PlcAddress](#).

C#

```
public bool Equals(PlcAddress other)
```

Parameters

[other](#) [PlcAddress](#)

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

Returns

[Boolean](#)

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

Exclusive()

Retrieves a new [PlcAddress](#) which points to the next address within the same [RawType](#) specific address sequence.

C#

```
public PlcAddress Exclusive()
```

Returns

[PlcAddress](#)

The [PlcAddress](#) exclusive the data area addressed by this [RawType](#).

Exclusive(PlcRawType)

Retrieves a new [PlcAddress](#) which points to the next address within the same [RawType](#) specific address sequence regarding the [rawType](#) specified.

C#

```
public PlcAddress Exclusive(PlcRawType rawType)
```

Parameters

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

The [PlcRawType](#) to be addressed by the new [PlcAddress](#).

Returns

[PlcAddress](#)

The [PlcAddress](#) exclusive the data area addressed by this [RawType](#).

GetHashCode()

Retrieves a hash code for this [PlcAddress](#).

C#

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

Returns

[Int32](#)

An [Int32](#) that contains the hash code for the [PlcAddress](#).

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).

Group(IEnumerable<PlcAddress>)

Groups the addresses of the sequence according to their [Operand](#).

C#

```
public static IDictionary<PlcOperand, IEnumerable<PlcAddress>> Group(IEnumerable<PlcAddress>
addresses)
```

Parameters

addresses [IEnumerable<PlcAddress>](#)

A sequence of [PlcAddress](#) instances whose elements to group.

Returns

[IDictionary<PlcOperand, IEnumerable>](#)

A dictionary where the key of an entry represents the group ([PlcOperand](#)) of one or more [PlcAddress](#) objects with the same [PlcOperand](#). These [PlcAddress](#) objects are stored as the value of the entry.

Exceptions

[ArgumentNullException](#)

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

Max(IEnumerable<PlcAddress>)

Returns the maximum address in a sequence of `PlcAddress` instances.

C#

```
public static PlcAddress Max(IEnumerable<PlcAddress> addresses)
```

Parameters

`addresses` `IEnumerable<PlcAddress>`

A sequence of `PlcAddress` instances to determine the maximum value of.

Returns

`PlcAddress`

The address that corresponds to the maximum value in the sequence.

Exceptions

`ArgumentNullException`

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

Max(PlcAddress[])

Returns the maximum address in a sequence of `PlcAddress` instances.

C#

```
public static PlcAddress Max(params PlcAddress[] addresses)
```

Parameters

`addresses` `PlcAddress[]`

A sequence of `PlcAddress` instances to determine the maximum value of.

Returns

`PlcAddress`

The address that corresponds to the maximum value in the sequence.

Exceptions

ArgumentNullException

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

Min(IEnumerable<PlcAddress>)

Returns the minimum address in a sequence of `PlcAddress` instances.

C#

```
public static PlcAddress Min(IEnumerable<PlcAddress> addresses)
```

Parameters

`addresses` `IEnumerable<PlcAddress>`

A sequence of `PlcAddress` instances to determine the minimum value of.

Returns

`PlcAddress`

The address that corresponds to the minimum value in the sequence.

Exceptions

`ArgumentNullException`

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

Min(PlcAddress[])

Returns the minimum address in a sequence of `PlcAddress` instances.

C#

```
public static PlcAddress Min(params PlcAddress[] addresses)
```

Parameters

`addresses` `PlcAddress[]`

A sequence of `PlcAddress` instances to determine the minimum value of.

Returns

`PlcAddress`

The address that corresponds to the minimum value in the sequence.

Exceptions

ArgumentNullException

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

Offset(Int32)

Creates and returns an adjusted copy of the `PlcAddress` class. The copy is adjusted by the specified amount. The original `PlcAddress` class remains unmodified.

C#

```
public PlcAddress Offset(int operandNumber)
```

Parameters

`operandNumber` `Int32`

The offset to use to adjust the `OperandNumber`.

Returns

`PlcAddress`

A new `PlcAddress` adjusted by the specified amount.

Exceptions

ArgumentOutOfRangeException

The `operandNumber` offset result into a new value that would be out of the bounds defined by `MinOperandNumber` and `MaxOperandNumber`.

Remarks

If the `PlcOperandType` does restrict the access to the `OperandNumber` (in case there `PlcOperandType` is not equals to `DataBlock`) the `OperandNumber` remains unchanged. If the `PlcRawType` does restrict the access to the `BitNumber` (in case there `PlcRawType` is not equals to `Bit`) the `BitNumber` remains unchanged.

Offset(Int32, Int32)

Creates and returns an adjusted copy of the `PlcAddress` class. The copy is adjusted by the specified amount. The original `PlcAddress` class remains unmodified.

C#

```
public PlcAddress Offset(int operandNumber, int byteNumber)
```


Parameters

operandNumber [Int32](#)

The offset to use to adjust the [OperandNumber](#).

byteNumber [Int32](#)

The offset to use to adjust the [ByteNumber](#).

Returns

[PlcAddress](#)

A new [PlcAddress](#) adjusted by the specified amount.

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](#) or [MaxByteNumber](#).

Remarks

If the [PlcOperandType](#) does restrict the access to the [OperandNumber](#) (in case there [PlcOperandType](#) is not equals to [DataBlock](#)) the [OperandNumber](#) remains unchanged. If the [PlcRawType](#) does restrict the access to the [BitNumber](#) (in case there [PlcRawType](#) is not equals to [Bit](#)) the [BitNumber](#) remains unchanged.

Offset(Int32, Int32, Int32)

Creates and returns an adjusted copy of the [PlcAddress](#) class. The copy is adjusted by the specified amount. The original [PlcAddress](#) class remains unmodified.

C#

```
public PlcAddress Offset(int operandNumber, int byteNumber, int bitNumber)
```

Parameters

operandNumber [Int32](#)

The offset to use to adjust the [OperandNumber](#).

byteNumber [Int32](#)

The offset to use to adjust the [ByteNumber](#).

bitNumber [Int32](#)

The offset to use to adjust the [BitNumber](#).

Returns

PlcAddress

A new [PlcAddress](#) adjusted by the specified amount.

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](#).

Remarks

If the [PlcOperandType](#) does restrict the access to the [OperandNumber](#) (in case there [PlcOperandType](#) is not equals to [DataBlock](#)) the [OperandNumber](#) remains unchanged. If the [PlcRawType](#) does restrict the access to the [BitNumber](#) (in case there [PlcRawType](#) is not equals to [Bit](#)) the [BitNumber](#) remains unchanged.

Parse(String)

Converts an address string to a [PlcAddress](#) instance.

C#

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

Parameters

[value](#) [String](#)

A string that contains an address.

Returns

PlcAddress

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

Exceptions

FormatException

The [value](#) is not a valid address.

ToString()

Converts the address to its string representation.

C#

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

Returns

String

A string that contains the address.

ToString(PlcOperandStandard)

Converts the address 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 address.

Returns

String

A string that contains the address.

ToString(PlcOperandStandard, PlcAddressFormat)

Converts the address to its string representation using the specified **standard** and **format**.

C#

```
public string ToString(PlcOperandStandard standard, PlcAddressFormat format)
```

Parameters

standard PlcOperandStandard

The **PlcOperandStandard** to use to identify the standard dependent characters in the string representation of the address.

format PlcAddressFormat

The **PlcAddressFormat** to use to format the address data.

Returns

String

A string that contains the address.

TryParse(String, out PlcAddress)

Determines whether a string is a valid address.

C#

```
public static bool TryParse(string value, out PlcAddress address)
```

Parameters

value String

The string to validate.

address PlcAddress

The PlcAddress version of the string.

Returns

Boolean

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

Operators

Equality(PlcAddress, PlcAddress)

Returns a value indicating whether two instance of PlcAddress are equal.

C#

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

GreaterThan(PlcAddress, PlcAddress)

Determines whether the first specified PlcAddress object is greater than the second specified PlcAddress object.

C#

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

GreaterThanOrEqual(PlcAddress, PlcAddress)

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

C#

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

Implicit(String to PlcAddress)

Converts a string formatted as address to an [PlcAddress](#) object.

C#

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

Exceptions

[FormatException](#)

The [value](#) is not a valid PLC address.

Inequality(PlcAddress, PlcAddress)

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

C#

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

LessThan(PlcAddress, PlcAddress)

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

C#

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

Exceptions

[ArgumentNullException](#)

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

LessThanOrEqual(PlcAddress, PlcAddress)

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

C#

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

Exceptions

[ArgumentNullException](#)

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

Table of Contents

Constructors	1
PlcAddress(PlcOperand, PlcRawType, Int32)	1
PlcAddress(PlcOperand, PlcRawType, Int32, Int32)	1
PlcAddress(PlcOperandType, Int32, PlcRawType, Int32)	2
PlcAddress(PlcOperandType, Int32, PlcRawType, Int32, Int32)	3
PlcAddress(PlcOperandType, PlcRawType, Int32)	4
PlcAddress(PlcOperandType, PlcRawType, Int32, Int32)	4
PlcAddress(SerializationInfo, StreamingContext)	5
Fields	6
MaxBitNumber	6
MaxByteNumber	6
MaxOperandNumber	6
MinBitNumber	6
MinByteNumber	7
MinOperandNumber	7
Properties	7
BitNumber	7
ByteNumber	7
Operand	8
OperandNumber	8
OperandType	8
OriginalString	9
RawType	9
Standard	9
Methods	9
CompareTo(Object)	9
CompareTo(PlcAddress)	10
Distinct(IEnumerable<PlcAddress>)	10
Equals(Object)	11
Equals(PlcAddress)	11
Exclusive()	12
Exclusive(PlcRawType)	12
GetHashCode()	12
GetObjectData(SerializationInfo, StreamingContext)	13
Group(IEnumerable<PlcAddress>)	13
Max(IEnumerable<PlcAddress>)	14
Max(PlcAddress[])	14
Min(IEnumerable<PlcAddress>)	15
Min(PlcAddress[])	15
Offset(Int32)	16
Offset(Int32, Int32)	16
Offset(Int32, Int32, Int32)	17
Parse(String)	18
ToString()	19
ToString(PlcOperandStandard)	19
ToString(PlcOperandStandard, PlcAddressFormat)	19
TryParse(String, out PlcAddress)	20
Operators	20
Equality(PlcAddress, PlcAddress)	20
GreaterThan(PlcAddress, PlcAddress)	20
GreaterThanOrEqual(PlcAddress, PlcAddress)	21

Implicit(String to PlcAddress)	21
Inequality(PlcAddress, PlcAddress)	21
LessThan(PlcAddress, PlcAddress)	21
LessThanOrEqual(PlcAddress, PlcAddress)	21