

PlcType Members

Namespace: IPS7Lnk.Advanced

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

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

Fields

MaxLength

Specifies the maximum value that can be used for the [Length](#) property.

C#

```
public const int MaxLength = 65535
```

Field Value

[Int32](#)

MinLength

Specifies the minimum value that can be used for the [Length](#) property.

C#

```
public const int MinLength =
```

Field Value

[Int32](#)

Properties

Address

Gets the start address of the [PlcType](#) in the PLC.

C#

```
public PlcAddress Address { get; }
```

Property Value

[PlcAddress](#)

The [PlcAddress](#) used as the start address of the [PlcType](#) at which data of the type represented is stored

within the PLC.

Remarks

The value of this property is a null reference (Nothing in Visual Basic) in cases there the [PlcType](#) is relative (see [IsAbsolute](#)).

DefaultValue

Gets the type dependent default value which can be assigned to an entity of the type represented.

C#

```
public virtual object DefaultValue { get; }
```

Property Value

[Object](#)

The type dependent default value.

Description

Gets or sets more meaningful information about the type 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 type and its usage.

EndAddress

Gets the end address of the [PlcType](#) in the PLC.

C#

```
public PlcAddress EndAddress { get; }
```

Property Value

[PlcAddress](#)

The [PlcAddress](#) used as the end address of the [PlcType](#) at which the data associated with the type represented ends within the PLC.

Remarks

The value of this property is a null reference (Nothing in Visual Basic) in cases there the [PlcType](#) is relative (see [IsAbsolute](#)).

The [PlcAddress](#) provided by this property does represent the inclusive address of the type represented.

FrameworkType

Gets the concrete .NET framework type used when working with the [PlcType](#).

C#

```
public Type FrameworkType { get; }
```

Property Value

[Type](#)

The concrete [Type](#) used when working with the [PlcType](#).

HasElementType

Gets a value indicating whether the current [PlcType](#) encompasses another type this is the case when the current [PlcType](#) is an array.

C#

```
public bool HasElementType { get; }
```

Property Value

[Boolean](#)

The value true if the [PlcType](#) is an array; otherwise the value false.

Identity

Gets the identity of the [PlcType](#) in the PLC.

C#

```
public PlcIdentity Identity { get; }
```

Property Value

[PlcIdentity](#)

The [PlcIdentity](#) used to identify the PLC resource associated with the [PlcType](#).

IsAbsolute

Gets a value indicating whether the [PlcType](#) does absolutely refer to a PLC resource.

C#

```
public bool IsAbsolute { get; }
```

Property Value

[Boolean](#)

The value true, if the [PlcType](#) does refer by its [Identity](#) to a absolute PLC resource; otherwise the value false.

Remarks

If [Address](#) isn't a null reference (Nothing in Visual Basic) in general [IsAbsolute](#) will always return the value true. In cases there the [Identity](#) is an instance of the [PlcOperand](#) class the property will return also the value true if the [Type](#) is equals [DataBlock](#).

IsArray

Gets a value indicating whether the framework [Type](#) of the [PlcType](#) is an array.

C#

```
public bool IsArray { get; }
```

Property Value

[Boolean](#)

The value true, if the framework [Type](#) is an array; otherwise the value false.

IsClass

Gets a value indicating whether the framework [Type](#) of the [PlcType](#) is a class; that is, not a value type or interface.

C#

```
public bool IsClass { get; }
```

Property Value

[Boolean](#)

The value true, if the framework [Type](#) is a class; otherwise the value false.

Layout

Gets the characteristics to use when aligning the [Members](#) of the [PlcType](#).

C#

```
public PlcTypeLayout Layout { get; }
```

Property Value

[PlcTypeLayout](#)

An instance of the [PlcTypeLayout](#) class to align the [Members](#).

Length

Gets a 32-bit integer that represents the total number of elements that can be stored in a value of the type.

C#

```
public int Length { get; }
```

Property Value

[Int32](#)

A 32-bit integer that represents the total number of elements that can be stored in a value of the type.

Members

Gets the members associated with the PLC type.

C#

```
public IPlcMemberInfo[] Members { get; }
```

Property Value

[IPlcMemberInfo\[\]](#)

An array of [IPlcMemberInfo](#) objects.

Name

Gets the name of the PLC type represented.

C#

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

Property Value

PlcName

The name of the PLC type.

RelativeType

Gets the [PlcType](#) from that this [PlcType](#) has been absolutely defined.

C#

```
public PlcType RelativeType { get; }
```

Property Value

PlcType

An instance of the [PlcType](#) class, if this [PlcType](#) has been originally defined by a non-absolute type declaration.

Size

Gets a 32-bit integer that represents the total number of bytes required to store a value of the type within the PLC memory.

C#

```
public int Size { get; }
```

Property Value

Int32

A 32-bit integer that represents the total number of bytes required to store a value of the type within the PLC memory.

SizeInBits

Gets a 32-bit integer that represents the total number of bits required to store a value of the type within the PLC memory.

C#

```
public int SizeInBits { get; }
```

Property Value

Int32

A 32-bit integer that represents the total number of bits required to store a value of the type within the PLC memory.

TypeCode

Gets the [PlcTypeCode](#) of the PLC type represented.

C#

```
public PlcTypeCode TypeCode { get; }
```

Property Value

[PlcTypeCode](#)

One of members defined by the [PlcTypeCode](#) enumeration. Which defines the concrete type representation in a PLC device.

Methods

CreateEntity()

Creates a new [IPlcEntity](#) object which represents a concrete accessible entity of this [PlcType](#).

C#

```
public virtual IPlcEntity CreateEntity()
```

Returns

[IPlcEntity](#)

An instance implementing the [IPlcEntity](#) interface its [Type](#) property does refer to this [PlcType](#).

Exceptions

[NotSupportedException](#)

This pure [PlcType](#) instance can't create an entity, because it is an abstract type representation.

Equals(Object)

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

C#

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

Parameters

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

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

Returns

[Boolean](#)

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

Equals(PlcType)

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

C#

```
public bool Equals(PlcType other)
```

Parameters

[other](#) [PlcType](#)

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

Returns

[Boolean](#)

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

GetElementType(Int32)

Retrieves the [PlcType](#) object encompassed or referred to by the current array type.

C#

```
public PlcType GetElementType(int index)
```

Parameters

[index](#) [Int32](#)

The zero based index of the element type to requested.

Returns

[PlcType](#)

The [PlcType](#) of the object encompassed or referred to by the current array or a null reference (Nothing in Visual Basic) if the current [PlcType](#) is not an array type.

Exceptions

ArgumentOutOfRangeException

The `index` is outside the bounds defined by the `MinLength` and `Length`.

GetHashCode()

Retrieves a hash code for this `PlcType`.

C#

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

Returns

`Int32`

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

GetRawType(PlcTypeCode)

Retrieves the according `PlcRawType` for the `typeCode` specified.

C#

```
public static PlcRawType GetRawType(PlcTypeCode typeCode)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` for that the `PlcRawType` is to be determined.

Returns

`PlcRawType`

The according `PlcRawType` for `typeCode`.

GetRawType(Type)

Retrieves the according `PlcRawType` for the `type` specified.

C#

```
public static PlcRawType GetRawType(Type type)
```

Parameters

`type` `Type`

The `Type` for that the `PlcRawType` is to be determined.

Returns

PlcRawType

The according `PlcRawType` for `type`.

Exceptions

ArgumentNullException

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

GetSize(PlcRawType)

Retrieves the according size for the `rawType` specified.

C#

```
public static int GetSize(PlcRawType rawType)
```

Parameters

`rawType` `PlcRawType`

The `PlcRawType` for that the size is to be determined.

Returns

Int32

The according size (in bytes) for `rawType`.

GetSize(PlcRawType, Boolean)

Retrieves the according size (optionally in bytes or bits) for the `rawType` specified.

C#

```
public static int GetSize(PlcRawType rawType, bool inBytes)
```

Parameters

`rawType` `PlcRawType`

The `PlcRawType` for that the size is to be determined.

`inBytes` `Boolean`

The value true, to retrieve the size of `rawType` in bytes; otherwise the value false to retrieve the size in bits.

Returns

Int32

The according size either in bytes or in bits for `rawType`.

GetSize(PlcTypeCode)

Retrieves the according size for the `typeCode` specified.

C#

```
public static int GetSize(PlcTypeCode typeCode)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcRawType` for that the size is to be determined.

Returns

Int32

The according size (in bytes) for `typeCode`.

GetSize(PlcTypeCode, Boolean)

Retrieves the according size (optionally in bytes or bits) for the `typeCode` specified.

C#

```
public static int GetSize(PlcTypeCode typeCode, bool inBytes)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` for that the size is to be determined.

`inBytes` `Boolean`

The value true, to retrieve the size of `typeCode` in bytes; otherwise the value false to retrieve the size in bits.

Returns

Int32

The according size either in bytes or in bits for `typeCode`.

GetSize(Type)

Retrieves the according size for the **type** specified.

C#

```
public static int GetSize(Type type)
```

Parameters

type Type

The **Type** for that the size is to be determined.

Returns

Int32

The according size (in bytes) for **type**.

Exceptions

ArgumentNullException

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

GetSize(Type, Boolean)

Retrieves the according size (optionally in bytes or bits) for the **type** specified.

C#

```
public static int GetSize(Type type, bool inBytes)
```

Parameters

type Type

The **Type** for that the size is to be determined.

inBytes Boolean

The value true, to retrieve the size of **type** in bytes; otherwise the value false to retrieve the size in bits.

Returns

Int32

The according size either in bytes or in bits for **type**.

Exceptions

ArgumentNullException

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

GetSupportedRawTypes(PlcTypeCode)

Retrieves the `PlcRawType` members supported by the `PlcTypeCode` specified by `typeCode`.

C#

```
public static PlcRawType[] GetSupportedRawTypes(PlcTypeCode typeCode)
```

Parameters

`typeCode` `PlcTypeCode`

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

Returns

`PlcRawType[]`

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

Remarks

Supported raw types means that these `PlcRawType` members can be used to address the PLC memory of such a `PlcTypeCode` by using one of the supported `PlcRawType` members.

GetSupportedRawTypes(Type)

Retrieves the `PlcRawType` members supported by the `PlcTypeCode` of the `Type` specified by `type`.

C#

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

Parameters

`type` `Type`

The `Type` its `PlcTypeCode` is to be used to determine the supported `PlcRawType` members.

Returns

`PlcRawType[]`

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

Exceptions

ArgumentNullException

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

Remarks

Supported raw types means that these `PlcRawType` members can be used to address the PLC memory of such a `PlcTypeCode` (identified by `type`) by using one of the supported `PlcRawType` members.

GetType(PlcTypeCode)

Retrieves the according framework `Type` for the `typeCode` specified.

C#

```
public static Type GetType(PlcTypeCode typeCode)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` for that the framework `Type` is to be determined.

Returns

`Type`

The according framework `Type` for `typeCode`.

Exceptions

`ArgumentException`

The `typeCode` is not supported.

GetTypeCode(Type)

Retrieves the according `PlcTypeCode` for the framework `Type` specified.

C#

```
public static PlcTypeCode GetTypeCode(Type type)
```

Parameters

`type` `Type`

The framework `Type` for that the `PlcTypeCode` is to be determined.

Returns

PlcTypeCode

The according **PlcTypeCode** for **type**.

Exceptions

ArgumentException

The **type** is not supported.

ArgumentNullException

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

IsSupported(PlcTypeCode, PlcRawType)

Determines whether the specified **rawType** can be used to address the PLC memory to access data of the **PlcTypeCode** specified by **typeCode**.

C#

```
public static bool IsSupported(PlcTypeCode typeCode, PlcRawType rawType)
```

Parameters

typeCode **PlcTypeCode**

The **PlcTypeCode** its supported raw types are inspected to determine the accessibility of the PLC memory for such PLC data using the **PlcRawType** specified by **rawType**.

rawType **PlcRawType**

The **PlcRawType** its compatibility to address PLC data of the **PlcTypeCode** defined by **typeCode** is to be inspected.

Returns

Boolean

The value true if the PLC data of the **typeCode** can be accessed using the **PlcRawType** specified by **rawType**; otherwise the value false.

IsSupportedArrayType(PlcTypeCode)

Determines whether the **PlcTypeCode** specified by **typeCode** can be also used to access the PLC memory by an array of that **PlcTypeCode**.

C#

```
public static bool IsSupportedArrayType(PlcTypeCode typeCode)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` to inspect.

Returns

`Boolean`

The value true if the PLC memory can be accessed by an array of the `PlcTypeCode` specified by `typeCode`; otherwise the value false.

MakeAbsoluteType(PlcIdentity)

Retrieves the absolute `PlcType` using the `PlcIdentity` specified by `identity`.

C#

```
public PlcType MakeAbsoluteType(PlcIdentity identity)
```

Parameters

`identity` `PlcIdentity`

The `PlcIdentity` to use when the current `PlcType` is a relative type and the absolute type of the `PlcType` is created.

Returns

`PlcType`

The reference to the current `PlcType` if `IsAbsolute` is equals true; otherwise another (but absolute) `PlcType` with the same type metadata including the `identity` specified.

Exceptions

`ArgumentException`

The `identity` specified does not provide the necessary information required to address the type of data.

`ArgumentNullException`

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

MakeArrayType()

Returns a `PlcType` object representing a one-dimensional array of the current type, with a length of zero.

C#

```
public PlcType MakeArrayType()
```

Returns

PlcType

A [PlcType](#) object representing a one-dimensional array of the current type, with a length of zero.

Exceptions

NotSupportedException

The current [PlcType](#) is either already an array type (see [IsArray](#)) which would result into an unsupported jagged array type or the current [PlcType](#) is a PLC type which does not support array types (see [IsSupportedArrayType\(PlcTypeCode\)](#)).

MakeArrayType(Int32)

Returns a [PlcType](#) object representing an array of the current type, with the specified [length](#).

C#

```
public PlcType MakeArrayType(int length)
```

Parameters

length Int32

The length for the array.

Returns

PlcType

An object representing an array of the current type, with the specified [length](#).

Exceptions

ArgumentOutOfRangeException

The [length](#) is out of the bounds defined by [MinLength](#) and [MaxLength](#).

NotSupportedException

The current [PlcType](#) is either already an array type (see [IsArray](#)) which would result into an unsupported jagged array type or the current [PlcType](#) is a PLC type which does not support array types (see [IsSupportedArrayType\(PlcTypeCode\)](#)).

Of(PlcTypeCode)

Retrieves the according [PlcType](#) which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode)
```

Parameters

typeCode [PlcTypeCode](#)

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

Returns

[PlcType](#)

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

[ArgumentException](#)

The **typeCode** is [Empty](#).

Remarks

Using this method it is possible to create from all [PlcTypeCode](#) member the according [PlcType](#) for value and object types. In general it is not possible to define [PlcType](#) which represent an array type using this method.

Of(PlcTypeCode, PlcIdentity)

Retrieves the according [PlcType](#) which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, PlcIdentity identity)
```

Parameters

typeCode [PlcTypeCode](#)

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

identity [PlcIdentity](#)

The [PlcIdentity](#) to which the [PlcType](#) does refer.

Returns

[PlcType](#)

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

ArgumentException

The `typeCode` is `Empty` or the `identity` can not be used together with the `typeCode` specified or `identity` does not provide the necessary information required to address the type of data.

ArgumentNullException

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

Remarks

Using this method it is possible to create from all `PlcTypeCode` member the according `PlcType` for value and object types. In general it is not possible to define `PlcType` which represent an array type using this method.

Of(PlcTypeCode, PlcIdentity, Int32)

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, PlcIdentity identity, int length)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` of the PLC type represented by the `PlcType`.

`identity` `PlcIdentity`

The `PlcIdentity` to which the `PlcType` does refer.

`length` `Int32`

The length of the `PlcType` which either specifies the total number of elements in an array type or the total number of characters in a string type.

Returns

`PlcType`

The `PlcType` which matches with the characteristics supplied.

Exceptions

ArgumentException

The `typeCode` is `Empty` or it can not be used together with the `identity` specified or `identity` does not provide the necessary information required to address the type of data.

ArgumentNullException

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

ArgumentOutOfRangeException

The `length` is out of the bounds defined by `MinLength` and `MaxLength` or `MinLength` and `MaxLength`.

Remarks

This method does always construct array types using the `PlcType`. Therefore it is not possible to use the following `PlcTypeCode` members as the value for `typeCode`: * `Date`.

- `Time`.
- `TimeOfDay`.
- `S5Time`.
- `DateTime`.
- `Object`.

It is also possible to construct a `PlcType` with the `String` through this method, there `length` does specify the maximum number of characters in the string.

Of(PlcTypeCode, PlcIdentity, PlcName)

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, PlcIdentity identity, PlcName name)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` of the PLC type represented by the `PlcType`.

`identity` `PlcIdentity`

The `PlcIdentity` to which the `PlcType` does refer.

`name` `PlcName`

The `PlcName` of the `PlcType`.

Returns

`PlcType`

The `PlcType` which matches with the characteristics supplied.

Exceptions

ArgumentException

The `typeCode` is `Empty` or the `identity` can not be used together with the `typeCode` specified or `identity`

does not provide the necessary information required to address the type of data.

ArgumentNullException

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

Remarks

Using this method it is possible to create from all `PlcTypeCode` member the according `PlcType` for value and object types. In general it is not possible to define `PlcType` which represent an array type using this method.

Of(PlcTypeCode, PlcIdentity, PlcName, Int32)

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, PlcIdentity identity, PlcName name, int length)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` of the PLC type represented by the `PlcType`.

`identity` `PlcIdentity`

The `PlcIdentity` to which the `PlcType` does refer.

`name` `PlcName`

The `PlcName` of the `PlcType`.

`length` `Int32`

The length of the `PlcType` which either specifies the total number of elements in an array type or the total number of characters in a string type.

Returns

`PlcType`

The `PlcType` which matches with the characteristics supplied.

Exceptions

`ArgumentException`

The `typeCode` is `Empty` or it can not be used together with the `identity` specified or `identity` does not provide the necessary information required to address the type of data.

`ArgumentNullException`

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

ArgumentOutOfRangeException

The `length` is out of the bounds defined by `MinLength` and `MaxLength` or `MinLength` and `MaxLength`.

Remarks

This method does always construct array types using the `PlcType`. Therefore it is not possible to use the following `PlcTypeCode` members as the value for `typeCode`: * `Date`.

- `Time`.
- `TimeOfDay`.
- `S5Time`.
- `DateTime`.
- `Object`.

It is also possible to construct a `PlcType` with the `String` through this method, there `length` does specify the maximum number of characters in the string.

Of(PlcTypeCode, PlcIdentity, PlcName, PlcTypeLayout, IEnumerable<IPlcMemberInfo>)

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, PlcIdentity identity, PlcName name,
    PlcTypeLayout layout, IEnumerable<IPlcMemberInfo> members)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` of the PLC type represented by the `PlcType`.

`identity` `PlcIdentity`

The `PlcIdentity` to which the `PlcType` does refer.

`name` `PlcName`

The `PlcName` of the `PlcType`.

`layout` `PlcTypeLayout`

An instance of the `PlcTypeLayout` class which provides the characteristics to use to absolutely align the `members`.

`members` `IEnumerable<IPlcMemberInfo>`

The `IPlcMemberInfo` objects to associate with the `PlcType`.

Returns

PlcType

The **PlcType** which matches with the characteristics supplied.

Exceptions

ArgumentException

The **typeCode** is **Empty** or the **identity** can not be used together with the **typeCode** specified or **identity** does not provide the necessary information required to address the type of data.

ArgumentNullException

The **identity** or **name**, **layout** or **members** or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct **PlcType** objects there the **typeCode** is equals **Object**.

Of(PlcTypeCode, PlcIdentity, PlcName, PlcTypeLayout, IPlcMemberInfo[])

Retrieves the according **PlcType** which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, PlcIdentity identity, PlcName name,
    PlcTypeLayout layout, params IPlcMemberInfo[] members)
```

Parameters

typeCode **PlcTypeCode**

The **PlcTypeCode** of the PLC type represented by the **PlcType**.

identity **PlcIdentity**

The **PlcIdentity** to which the **PlcType** does refer.

name **PlcName**

The **PlcName** of the **PlcType**.

layout **PlcTypeLayout**

An instance of the **PlcTypeLayout** class which provides the characteristics to use to absolutely align the **members**.

members **IPlcMemberInfo[]**

The **IPlcMemberInfo** objects to associate with the **PlcType**.

Returns

PlcType

The **PlcType** which matches with the characteristics supplied.

Exceptions

ArgumentException

The **typeCode** is **Empty** or the **identity** can not be used together with the **typeCode** specified or **identity** does not provide the necessary information required to address the type of data.

ArgumentNullException

The **identity**, **name**, **layout** or **members** or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct **PlcType** objects there the **typeCode** is equals **Object**.

Of(PlcTypeCode, PlcIdentity, PlcTypeLayout, IEnumerable<IPlcMemberInfo>)

Retrieves the according **PlcType** which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, PlcIdentity identity, PlcTypeLayout layout,
    IEnumerable<IPlcMemberInfo> members)
```

Parameters

typeCode **PlcTypeCode**

The **PlcTypeCode** of the PLC type represented by the **PlcType**.

identity **PlcIdentity**

The **PlcIdentity** to which the **PlcType** does refer.

layout **PlcTypeLayout**

An instance of the **PlcTypeLayout** class which provides the characteristics to use to absolutely align the **members**.

members **IEnumerable<IPlcMemberInfo>**

The **IPlcMemberInfo** objects to associate with the **PlcType**.

Returns

PlcType

The **PlcType** which matches with the characteristics supplied.

Exceptions

ArgumentException

The `typeCode` is `Empty` or the `identity` can not be used together with the `typeCode` specified or `identity` does not provide the necessary information required to address the type of data.

ArgumentNullException

The `identity`, `layout` or `members` or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct `PlcType` objects there the `typeCode` is equals `Object`.

Of(PlcTypeCode, PlcIdentity, PlcTypeLayout, IPlcMemberInfo[])

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, PlcIdentity identity, PlcTypeLayout layout,
    params IPlcMemberInfo[] members)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` of the PLC type represented by the `PlcType`.

`identity` `PlcIdentity`

The `PlcIdentity` to which the `PlcType` does refer.

`layout` `PlcTypeLayout`

An instance of the `PlcTypeLayout` class which provides the characteristics to use to absolutely align the `members`.

`members` `IPlcMemberInfo[]`

The `IPlcMemberInfo` objects to associate with the `PlcType`.

Returns

`PlcType`

The `PlcType` which matches with the characteristics supplied.

Exceptions

ArgumentException

The `typeCode` is `Empty` or the `identity` can not be used together with the `typeCode` specified or `identity` does not provide the necessary information required to address the type of data.

ArgumentNullException

The `identity`, `layout` or `members` or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct `PlcType` objects there the `typeCode` is equals `Object`.

Of(PlcTypeCode, PlcTypeLayout, IEnumerable<IPlcMemberInfo>)

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, PlcTypeLayout layout,
IEnumerable<IPlcMemberInfo> members)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` of the PLC type represented by the `PlcType`.

`layout` `PlcTypeLayout`

An instance of the `PlcTypeLayout` class which provides the characteristics to use to absolutely align the `members`.

`members` `IEnumerable<IPlcMemberInfo>`

The `IPlcMemberInfo` objects to associate with the `PlcType`.

Returns

`PlcType`

The `PlcType` which matches with the characteristics supplied.

Exceptions

`ArgumentException`

The `typeCode` is `Empty`.

`ArgumentNullException`

The `layout` or `members` or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct [PlcType](#) objects there the [typeCode](#) is equals [Object](#).

Of(PlcTypeCode, PlcTypeLayout, IPlcMemberInfo[])

Retrieves the according [PlcType](#) which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, PlcTypeLayout layout, params IPlcMemberInfo[] members)
```

Parameters

[typeCode](#) [PlcTypeCode](#)

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

[layout](#) [PlcTypeLayout](#)

An instance of the [PlcTypeLayout](#) class which provides the characteristics to use to absolutely align the [members](#).

[members](#) [IPlcMemberInfo\[\]](#)

The [IPlcMemberInfo](#) objects to associate with the [PlcType](#).

Returns

[PlcType](#)

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

[ArgumentException](#)

The [typeCode](#) is [Empty](#).

[ArgumentNullException](#)

The [layout](#) or [members](#) or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct [PlcType](#) objects there the [typeCode](#) is equals [Object](#).

Of(PlcTypeCode, Type)

Retrieves the according [PlcType](#) which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType)
```

Parameters

typeCode [PlcTypeCode](#)

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

frameworkType [Type](#)

The framework [Type](#) represented by the [PlcType](#).

Returns

[PlcType](#)

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

[ArgumentException](#)

The **typeCode** is [Empty](#) or **frameworkType** is a not supported framework type or a jagged or multi-dimensional array type.

[ArgumentNullException](#)

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

Of(PlcTypeCode, Type, Int32)

Retrieves the according [PlcType](#) which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType, int length)
```

Parameters

typeCode [PlcTypeCode](#)

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

frameworkType [Type](#)

The framework [Type](#) represented by the [PlcType](#).

length [Int32](#)

The length of the [PlcType](#) which either specifies the total number of elements in an array type or the total number of characters in a string type.

Returns

[PlcType](#)

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

ArgumentException

The `typeCode` is `Empty` or `frameworkType` is a not supported framework type or a jagged or multi-dimensional array type.

ArgumentNullException

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

ArgumentOutOfRangeException

The `Length` is out of the bounds defined by `MinLength` and `MaxLength` or `MinLength` and `MaxLength`.

Of(PlcTypeCode, Type, Int32, PlcTypeLayout, IEnumerable<IPlcMemberInfo>)

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType, int length, PlcTypeLayout layout, IEnumerable<IPlcMemberInfo> members)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` of the PLC type represented by the `PlcType`.

`frameworkType` `Type`

The framework `Type` represented by the `PlcType`.

`Length` `Int32`

The length of the `PlcType` which either specifies the total number of elements in an array type or the total number of characters in a string type.

`Layout` `PlcTypeLayout`

An instance of the `PlcTypeLayout` class which provides the characteristics to use to absolutely align the `members`.

`members` `IEnumerable<IPlcMemberInfo>`

The `IPlcMemberInfo` objects to associate with the `PlcType`.

Returns

`PlcType`

The `PlcType` which matches with the characteristics supplied.

Exceptions

ArgumentException

The `typeCode` is `Empty` or `frameworkType` is a not supported framework type or a jagged or multi-dimensional array type.

ArgumentNullException

The `frameworkType`, `layout` or `members` or one of its items is a null reference (Nothing in Visual Basic).

ArgumentOutOfRangeException

The `length` is out of the bounds defined by `MinLength` and `MaxLength` or `MinLength` and `MaxLength`.

Of(PlcTypeCode, Type, Int32, PlcTypeLayout, IPlcMemberInfo[])

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType, int length, PlcTypeLayout layout, params IPlcMemberInfo[] members)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` of the PLC type represented by the `PlcType`.

`frameworkType` `Type`

The framework `Type` represented by the `PlcType`.

`length` `Int32`

The length of the `PlcType` which either specifies the total number of elements in an array type or the total number of characters in a string type.

`layout` `PlcTypeLayout`

An instance of the `PlcTypeLayout` class which provides the characteristics to use to absolutely align the `members`.

`members` `IPlcMemberInfo[]`

The `IPlcMemberInfo` objects to associate with the `PlcType`.

Returns

`PlcType`

The `PlcType` which matches with the characteristics supplied.

Exceptions

ArgumentException

The `typeCode` is `Empty` or `frameworkType` is a not supported framework type or a jagged or multi-dimensional array type.

ArgumentNullException

The `frameworkType`, `layout` or `members` or one of its items is a null reference (Nothing in Visual Basic).

ArgumentOutOfRangeException

The `Length` is out of the bounds defined by `MinLength` and `MaxLength` or `MinLength` and `MaxLength`.

Of(PlcTypeCode, Type, PlcIdentity)

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType, PlcIdentity identity)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` of the PLC type represented by the `PlcType`.

`frameworkType` `Type`

The framework `Type` represented by the `PlcType`.

`identity` `PlcIdentity`

The `PlcIdentity` to which the `PlcType` does refer.

Returns

`PlcType`

The `PlcType` which matches with the characteristics supplied.

Exceptions

ArgumentException

The `typeCode` is `Empty` or `frameworkType` is a not supported framework type or a jagged or multi-dimensional array type or `typeCode` / `frameworkType` can not be used together with the `identity` specified or `identity` does not provide the necessary information required to address the type of data.

ArgumentNullException

The `frameworkType` or `identity` is a null reference (Nothing in Visual Basic).

Of(PlcTypeCode, Type, PlcIdentity, Int32)

Retrieves the according [PlcType](#) which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType, PlcIdentity identity, int length)
```

Parameters

typeCode [PlcTypeCode](#)

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

frameworkType [Type](#)

The framework [Type](#) represented by the [PlcType](#).

identity [PlcIdentity](#)

The [PlcIdentity](#) to which the [PlcType](#) does refer.

length [Int32](#)

The length of the [PlcType](#) which either specifies the total number of elements in an array type or the total number of characters in a string type.

Returns

[PlcType](#)

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

[ArgumentException](#)

The **typeCode** is [Empty](#) or **frameworkType** is a not supported framework type or a jagged or multi-dimensional array type or **typeCode** / **frameworkType** can not be used together with the **identity** specified or **identity** does not provide the necessary information required to address the type of data.

[ArgumentNullException](#)

The **frameworkType** or **identity** is a null reference (Nothing in Visual Basic).

[ArgumentOutOfRangeException](#)

The **length** is out of the bounds defined by [MinLength](#) and [MaxLength](#) or [MinLength](#) and [MaxLength](#).

Of(PlcTypeCode, Type, PlcIdentity, Int32, PlcTypeLayout, IEnumerable<IPlcMemberInfo>)

Retrieves the according [PlcType](#) which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType, PlcIdentity identity, int length, PlcTypeLayout layout, IEnumerable<IPlcMemberInfo> members)
```

Parameters

typeCode [PlcTypeCode](#)

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

frameworkType [Type](#)

The framework [Type](#) represented by the [PlcType](#).

identity [PlcIdentity](#)

The [PlcIdentity](#) to which the [PlcType](#) does refer.

length [Int32](#)

The length of the [PlcType](#) which either specifies the total number of elements in an array type or the total number of characters in a string type.

layout [PlcTypeLayout](#)

An instance of the [PlcTypeLayout](#) class which provides the characteristics to use to absolutely align the **members**.

members [IEnumerable<IPlcMemberInfo>](#)

The [IPlcMemberInfo](#) objects to associate with the [PlcType](#).

Returns

[PlcType](#)

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

[ArgumentException](#)

The **typeCode** is [Empty](#) or **frameworkType** is a not supported framework type or a jagged or multi-dimensional array type or **typeCode** / **frameworkType** can not be used together with the **identity** specified or **identity** does not provide the necessary information required to address the type of data.

[ArgumentNullException](#)

The **frameworkType**, **identity**, **layout** or **members** or one of its items is a null reference (Nothing in Visual Basic).

[ArgumentOutOfRangeException](#)

The **length** is out of the bounds defined by [MinLength](#) and [MaxLength](#) or [MinLength](#) and [MaxLength](#).

Of(PlcTypeCode, Type, PlcIdentity, Int32, PlcTypeLayout, IPlcMemberInfo[])

Retrieves the according [PlcType](#) which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType, PlcIdentity identity, int length, PlcTypeLayout layout, params IPlcMemberInfo[] members)
```

Parameters

typeCode [PlcTypeCode](#)

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

frameworkType [Type](#)

The framework [Type](#) represented by the [PlcType](#).

identity [PlcIdentity](#)

The [PlcIdentity](#) to which the [PlcType](#) does refer.

length [Int32](#)

The length of the [PlcType](#) which either specifies the total number of elements in an array type or the total number of characters in a string type.

layout [PlcTypeLayout](#)

An instance of the [PlcTypeLayout](#) class which provides the characteristics to use to absolutely align the **members**.

members [IPlcMemberInfo\[\]](#)

The [IPlcMemberInfo](#) objects to associate with the [PlcType](#).

Returns

[PlcType](#)

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

[ArgumentException](#)

The **typeCode** is [Empty](#) or **frameworkType** is a not supported framework type or a jagged or multi-dimensional array type or **typeCode** / **frameworkType** can not be used together with the **identity** specified or **identity** does not provide the necessary information required to address the type of data.

[ArgumentNullException](#)

The **frameworkType**, **identity**, **layout** or **members** or one of its items is a null reference (Nothing in Visual Basic).

ArgumentOutOfRangeException

The **Length** is out of the bounds defined by **MinLength** and **MaxLength** or **MinLength** and **MaxLength**.

Of(PlcTypeCode, Type, PlcIdentity, PlcName)

Retrieves the according **PlcType** which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType, PlcIdentity identity,
    PlcName name)
```

Parameters

typeCode **PlcTypeCode**

The **PlcTypeCode** of the PLC type represented by the **PlcType**.

frameworkType **Type**

The framework **Type** represented by the **PlcType**.

identity **PlcIdentity**

The **PlcIdentity** to which the **PlcType** does refer.

name **PlcName**

The **PlcName** of the **PlcType**.

Returns

PlcType

The **PlcType** which matches with the characteristics supplied.

Exceptions

ArgumentException

The **typeCode** is **Empty** or **frameworkType** is a not supported framework type or a jagged or multi-dimensional array type or **typeCode** / **frameworkType** can not be used together with the **identity** specified or **identity** does not provide the necessary information required to address the type of data.

ArgumentNullException

The **frameworkType**, **identity** or **name** is a null reference (Nothing in Visual Basic).

Of(PlcTypeCode, Type, PlcIdentity, PlcName, Int32)

Retrieves the according **PlcType** which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType, PlcIdentity identity,
    PlcName name, int length)
```

Parameters

typeCode [PlcTypeCode](#)

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

frameworkType [Type](#)

The framework [Type](#) represented by the [PlcType](#).

identity [PlcIdentity](#)

The [PlcIdentity](#) to which the [PlcType](#) does refer.

name [PlcName](#)

The [PlcName](#) of the [PlcType](#).

length [Int32](#)

The length of the [PlcType](#) which either specifies the total number of elements in an array type or the total number of characters in a string type.

Returns

[PlcType](#)

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

[ArgumentException](#)

The **typeCode** is [Empty](#) or **frameworkType** is a not supported framework type or a jagged or multi-dimensional array type or **typeCode** / **frameworkType** can not be used together with the **identity** specified or **identity** does not provide the necessary information required to address the type of data.

[ArgumentNullException](#)

The **frameworkType**, **identity** or **name** is a null reference (Nothing in Visual Basic).

[ArgumentOutOfRangeException](#)

The **length** is out of the bounds defined by [MinLength](#) and [MaxLength](#) or [MinLength](#) and [MaxLength](#).

Of(PlcTypeCode, Type, PlcIdentity, PlcName, Int32, PlcTypeLayout, IEnumerable<IPlcMemberInfo>)

Retrieves the according [PlcType](#) which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType, PlcIdentity identity,
PlcName name, int length, PlcTypeLayout layout, IEnumerable<IPlcMemberInfo> members)
```

Parameters

typeCode [PlcTypeCode](#)

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

frameworkType [Type](#)

The framework [Type](#) represented by the [PlcType](#).

identity [PlcIdentity](#)

The [PlcIdentity](#) to which the [PlcType](#) does refer.

name [PlcName](#)

The [PlcName](#) of the [PlcType](#).

length [Int32](#)

The length of the [PlcType](#) which either specifies the total number of elements in an array type or the total number of characters in a string type.

layout [PlcTypeLayout](#)

An instance of the [PlcTypeLayout](#) class which provides the characteristics to use to absolutely align the [members](#).

members [IEnumerable<IPlcMemberInfo>](#)

The [IPlcMemberInfo](#) objects to associate with the [PlcType](#).

Returns

[PlcType](#)

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

[ArgumentException](#)

The **typeCode** is [Empty](#) or **frameworkType** is a not supported framework type or a jagged or multi-dimensional array type or **typeCode** / **frameworkType** can not be used together with the **identity** specified or **identity** does not provide the necessary information required to address the type of data.

[ArgumentNullException](#)

The **frameworkType**, **identity**, **layout** or **members** or one of its items is a null reference (Nothing in Visual Basic).

[ArgumentOutOfRangeException](#)

The **length** is out of the bounds defined by [MinLength](#) and [MaxLength](#) or [MinLength](#) and [MaxLength](#).

Of(PlcTypeCode, Type, PlcIdentity, PlcName, Int32, PlcTypeLayout, IPlcMemberInfo[])

Retrieves the according [PlcType](#) which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType, PlcIdentity identity,
PlcName name, int length, PlcTypeLayout layout, params IPlcMemberInfo[] members)
```

Parameters

typeCode [PlcTypeCode](#)

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

frameworkType [Type](#)

The framework [Type](#) represented by the [PlcType](#).

identity [PlcIdentity](#)

The [PlcIdentity](#) to which the [PlcType](#) does refer.

name [PlcName](#)

The [PlcName](#) of the [PlcType](#).

length [Int32](#)

The length of the [PlcType](#) which either specifies the total number of elements in an array type or the total number of characters in a string type.

layout [PlcTypeLayout](#)

An instance of the [PlcTypeLayout](#) class which provides the characteristics to use to absolutely align the **members**.

members [IPlcMemberInfo\[\]](#)

The [IPlcMemberInfo](#) objects to associate with the [PlcType](#).

Returns

[PlcType](#)

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

[ArgumentException](#)

The **typeCode** is [Empty](#) or **frameworkType** is a not supported framework type or a jagged or multi-dimensional array type or **typeCode** / **frameworkType** can not be used together with the **identity** specified or **identity** does not provide the necessary information required to address the type of data.

ArgumentNullException

The `frameworkType`, `identity`, `name`, `layout` or `members` or one of its items is a null reference (Nothing in Visual Basic).

ArgumentOutOfRangeException

The `length` is out of the bounds defined by `MinLength` and `MaxLength` or `MinLength` and `MaxLength`.

Of(PlcTypeCode, Type, PlcIdentity, PlcName, PlcTypeLayout, IEnumerable<IPlcMemberInfo>)

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType, PlcIdentity identity,
    PlcName name, PlcTypeLayout layout, IEnumerable<IPlcMemberInfo> members)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` of the PLC type represented by the `PlcType`.

`frameworkType` `Type`

The framework `Type` represented by the `PlcType`.

`identity` `PlcIdentity`

The `PlcIdentity` to which the `PlcType` does refer.

`name` `PlcName`

The `PlcName` of the `PlcType`.

`layout` `PlcTypeLayout`

An instance of the `PlcTypeLayout` class which provides the characteristics to use to absolutely align the `members`.

`members` `IEnumerable<IPlcMemberInfo>`

The `IPlcMemberInfo` objects to associate with the `PlcType`.

Returns

`PlcType`

The `PlcType` which matches with the characteristics supplied.

Exceptions

`ArgumentException`

The `typeCode` is `Empty` or `frameworkType` is a not supported framework type or a jagged or multi-dimensional array type or `typeCode` / `frameworkType` can not be used together with the `identity` specified or `identity` does not provide the necessary information required to address the type of data.

ArgumentNullException

The `frameworkType`, `identity`, `name`, `layout` or `members` or one of its items is a null reference (Nothing in Visual Basic).

Of(PlcTypeCode, Type, PlcIdentity, PlcName, PlcTypeLayout, IPlcMemberInfo[])

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType, PlcIdentity identity,
    PlcName name, PlcTypeLayout layout, params IPlcMemberInfo[] members)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` of the PLC type represented by the `PlcType`.

`frameworkType` `Type`

The framework `Type` represented by the `PlcType`.

`identity` `PlcIdentity`

The `PlcIdentity` to which the `PlcType` does refer.

`name` `PlcName`

The `PlcName` of the `PlcType`.

`layout` `PlcTypeLayout`

An instance of the `PlcTypeLayout` class which provides the characteristics to use to absolutely align the `members`.

`members` `IPlcMemberInfo[]`

The `IPlcMemberInfo` objects to associate with the `PlcType`.

Returns

`PlcType`

The `PlcType` which matches with the characteristics supplied.

Exceptions

`ArgumentException`

The `typeCode` is `Empty` or `frameworkType` is a not supported framework type or a jagged or multi-dimensional array type or `typeCode` / `frameworkType` can not be used together with the `identity` specified or `identity` does not provide the necessary information required to address the type of data.

ArgumentNullException

The `frameworkType`, `identity`, `layout` or `name` or `members` or one of its items is a null reference (Nothing in Visual Basic).

Of(PlcTypeCode, Type, PlcIdentity, PlcTypeLayout, IEnumerable<IPlcMemberInfo>)

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType, PlcIdentity identity,
    PlcTypeLayout layout, IEnumerable<IPlcMemberInfo> members)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` of the PLC type represented by the `PlcType`.

`frameworkType` `Type`

The framework `Type` represented by the `PlcType`.

`identity` `PlcIdentity`

The `PlcIdentity` to which the `PlcType` does refer.

`layout` `PlcTypeLayout`

An instance of the `PlcTypeLayout` class which provides the characteristics to use to absolutely align the `members`.

`members` `IEnumerable<IPlcMemberInfo>`

The `IPlcMemberInfo` objects to associate with the `PlcType`.

Returns

`PlcType`

The `PlcType` which matches with the characteristics supplied.

Exceptions

ArgumentException

The `typeCode` is `Empty` or `frameworkType` is a not supported framework type or a jagged or multi-dimensional array type or `typeCode` / `frameworkType` can not be used together with the `identity`

specified or `identity` does not provide the necessary information required to address the type of data.

ArgumentNullException

The `frameworkType`, `identity`, `layout` or `members` or one of its items is a null reference (Nothing in Visual Basic).

Of(PlcTypeCode, Type, PlcIdentity, PlcTypeLayout, IPlcMemberInfo[])

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType, PlcIdentity identity,
    PlcTypeLayout layout, params IPlcMemberInfo[] members)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` of the PLC type represented by the `PlcType`.

`frameworkType` `Type`

The framework `Type` represented by the `PlcType`.

`identity` `PlcIdentity`

The `PlcIdentity` to which the `PlcType` does refer.

`layout` `PlcTypeLayout`

An instance of the `PlcTypeLayout` class which provides the characteristics to use to absolutely align the `members`.

`members` `IPlcMemberInfo[]`

The `IPlcMemberInfo` objects to associate with the `PlcType`.

Returns

`PlcType`

The `PlcType` which matches with the characteristics supplied.

Exceptions

ArgumentException

The `typeCode` is `Empty` or `frameworkType` is a not supported framework type or a jagged or multi-dimensional array type or `typeCode` / `frameworkType` can not be used together with the `identity` specified or `identity` does not provide the necessary information required to address the type of data.

ArgumentNullException

The `frameworkType`, `identity`, `layout` or `members` or one of its items is a null reference (Nothing in Visual Basic).

Of(PlcTypeCode, Type, PlcTypeLayout, IEnumerable<IPlcMemberInfo>)

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType, PlcTypeLayout layout,
    IEnumerable<IPlcMemberInfo> members)
```

Parameters

`typeCode` `PlcTypeCode`

The `PlcTypeCode` of the PLC type represented by the `PlcType`.

`frameworkType` `Type`

The framework `Type` represented by the `PlcType`.

`layout` `PlcTypeLayout`

An instance of the `PlcTypeLayout` class which provides the characteristics to use to absolutely align the `members`.

`members` `IEnumerable<IPlcMemberInfo>`

The `IPlcMemberInfo` objects to associate with the `PlcType`.

Returns

`PlcType`

The `PlcType` which matches with the characteristics supplied.

Exceptions

`ArgumentException`

The `typeCode` is `Empty` or `frameworkType` is a not supported framework type or a jagged or multi-dimensional array type.

`ArgumentNullException`

The `frameworkType`, `layout` or `members` or one of its items is a null reference (Nothing in Visual Basic).

Of(PlcTypeCode, Type, PlcTypeLayout, IPlcMemberInfo[])

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(PlcTypeCode typeCode, Type frameworkType, PlcTypeLayout layout,
    params IPlcMemberInfo[] members)
```

Parameters

typeCode [PlcTypeCode](#)

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

frameworkType [Type](#)

The framework [Type](#) represented by the [PlcType](#).

layout [PlcTypeLayout](#)

An instance of the [PlcTypeLayout](#) class which provides the characteristics to use to absolutely align the [members](#).

members [IPlcMemberInfo\[\]](#)

The [IPlcMemberInfo](#) objects to associate with the [PlcType](#).

Returns

[PlcType](#)

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

[ArgumentException](#)

The **typeCode** is [Empty](#) or **frameworkType** is a not supported framework type or a jagged or multi-dimensional array type.

[ArgumentNullException](#)

The **frameworkType**, **layout** or **members** or one of its items is a null reference (Nothing in Visual Basic).

Of(Type)

Retrieves the according [PlcType](#) which matches the supplied type metadata.

C#

```
public static PlcType Of(Type type)
```

Parameters

type [Type](#)

The framework [Type](#) represented by the [PlcType](#).

Returns

PlcType

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

ArgumentException

The [type](#) is a not supported framework type or a jagged multi-dimensional array type.

ArgumentNullException

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

Remarks

If [type](#) is a value type then [PlcType](#) will be also a value type. If [type](#) is an array type then [PlcType](#) will be a array type with the length equals zero. The following [PlcType](#) can not constructed using this method: * [Date](#).

- [Time](#).
- [S5Time](#).

In general specifying [type](#) as a [DateTime](#) will result into a [PlcType](#) with the [DateTime](#) and specifying [type](#) as a [TimeSpan](#) will result into a [PlcType](#) with the [TimeOfDay](#).

Of(Type, PlcIdentity)

Retrieves the according [PlcType](#) which matches the supplied type metadata.

C#

```
public static PlcType Of(Type type, PlcIdentity identity)
```

Parameters

type Type

The framework [Type](#) represented by the [PlcType](#).

identity PlcIdentity

The [PlcIdentity](#) to which the [PlcType](#) does refer.

Returns

PlcType

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

ArgumentException

The `type` is a not supported framework type or a jagged multi-dimensional array type or it can not be used together with the `identity` specified or `identity` does not provide the necessary information required to address the type of data.

ArgumentNullException

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

Remarks

If `type` is a value type then `PlcType` will be also a value type. If `type` is an array type then `PlcType` will be a array type with the length equals zero. The following `PlcType` can not constructed using this method: *
`Date`.

- `Time`.
- `S5Time`.

In general specifying `type` as a `DateTime` will result into a `PlcType` with the `DateTime` and specifying `type` as a `TimeSpan` will result into a `PlcType` with the `TimeOfDay`.

Of(Type, PlcIdentity, Int32)

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(Type type, PlcIdentity identity, int length)
```

Parameters

`type` Type

The framework `Type` represented by the `PlcType`.

`identity` PlcIdentity

The `PlcIdentity` to which the `PlcType` does refer.

`length` Int32

The length of the `PlcType` which either specifies the total number of elements in an array type or the total number of characters in a string type.

Returns

PlcType

The `PlcType` which matches with the characteristics supplied.

Exceptions

ArgumentException

The **type** is a not supported framework type or a jagged multi-dimensional array type or it can not be used together with the **identity** specified or **identity** does not provide the necessary information required to address the type of data.

ArgumentNullException

The **type** or **identity** is a null reference (Nothing in Visual Basic).

ArgumentOutOfRangeException

The **length** is out of the bounds defined by **MinLength** and **MaxLength** or **MinLength** and **MaxLength**.

Remarks

This method does always construct array types using the **PlcType**. Therefore it is not possible to use the following framework types as the value for **type**: * **DateTime**.

- **TimeSpan**.
- **PlcObject** or derivate from it.

It is also possible to construct a **PlcType** with the **String** through this method, there **length** does specify the maximum number of characters in the string.

Of(Type, PlcIdentity, PlcName)

Retrieves the according **PlcType** which matches the supplied type metadata.

C#

```
public static PlcType Of(Type type, PlcIdentity identity, PlcName name)
```

Parameters

type Type

The framework **Type** represented by the **PlcType**.

identity PlcIdentity

The **PlcIdentity** to which the **PlcType** does refer.

name PlcName

The **PlcName** of the **PlcType**.

Returns

PlcType

The **PlcType** which matches with the characteristics supplied.

Exceptions

ArgumentException

The `type` is a not supported framework type or a jagged multi-dimensional array type or it can not be used together with the `identity` specified or `identity` does not provide the necessary information required to address the type of data.

ArgumentNullException

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

Remarks

If `type` is a value type then `PlcType` will be also a value type. If `type` is an array type then `PlcType` will be a array type with the length equals zero. The following `PlcType` can not constructed using this method: *
`Date`.

- `Time`.
- `S5Time`.

In general specifying `type` as a `DateTime` will result into a `PlcType` with the `DateTime` and specifying `type` as a `TimeSpan` will result into a `PlcType` with the `TimeOfDay`.

Of(Type, PlcIdentity, PlcName, Int32)

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(Type type, PlcIdentity identity, PlcName name, int length)
```

Parameters

`type` `Type`

The framework `Type` represented by the `PlcType`.

`identity` `PlcIdentity`

The `PlcIdentity` to which the `PlcType` does refer.

`name` `PlcName`

The `PlcName` of the `PlcType`.

`length` `Int32`

The length of the `PlcType` which either specifies the total number of elements in an array type or the total number of characters in a string type.

Returns

`PlcType`

The `PlcType` which matches with the characteristics supplied.

Exceptions

ArgumentException

The `type` is a not supported framework type or a jagged multi-dimensional array type or it can not be used together with the `identity` specified or `identity` does not provide the necessary information required to address the type of data.

ArgumentNullException

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

ArgumentOutOfRangeException

The `length` is out of the bounds defined by `MinLength` and `MaxLength` or `MinLength` and `MaxLength`.

Remarks

This method does always construct array types using the `PlcType`. Therefore it is not possible to use the following framework types as the value for `type`: * `DateTime`.

- `TimeSpan`.
- `PlcObject` or derivate from it.

It is also possible to construct a `PlcType` with the `String` through this method, there `length` does specify the maximum number of characters in the string.

Of(Type, PlcIdentity, PlcName, PlcTypeLayout, IEnumerable<IPlcMemberInfo>)

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of(Type type, PlcIdentity identity, PlcName name, PlcTypeLayout layout, IEnumerable<IPlcMemberInfo> members)
```

Parameters

`type` `Type`

The framework `Type` represented by the `PlcType`.

`identity` `PlcIdentity`

The `PlcIdentity` to which the `PlcType` does refer.

`name` `PlcName`

The `PlcName` of the `PlcType`.

`layout` `PlcTypeLayout`

An instance of the `PlcTypeLayout` class which provides the characteristics to use to absolutely align the `members`.

`members` `IEnumerable<IPlcMemberInfo>`

The [IPlcMemberInfo](#) objects to associate with the [PlcType](#).

Returns

[PlcType](#)

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

[ArgumentException](#)

The [type](#) is a not supported framework type or a jagged multi-dimensional array type or it can not be used together with the [identity](#) specified or [identity](#) does not provide the necessary information required to address the type of data.

[ArgumentNullException](#)

The [type](#), [identity](#), [name](#), [layout](#) or [members](#) or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct [PlcType](#) objects there the [type](#) is either of the type of [PlcObject](#) or a derivate of the class.

Of(Type, PlcIdentity, PlcName, PlcTypeLayout, IPlcMemberInfo[])

Retrieves the according [PlcType](#) which matches the supplied type metadata.

C#

```
public static PlcType Of(Type type, PlcIdentity identity, PlcName name, PlcTypeLayout layout, params IPlcMemberInfo[] members)
```

Parameters

[type](#) [Type](#)

The framework [Type](#) represented by the [PlcType](#).

[identity](#) [PlcIdentity](#)

The [PlcIdentity](#) to which the [PlcType](#) does refer.

[name](#) [PlcName](#)

The [PlcName](#) of the [PlcType](#).

[layout](#) [PlcTypeLayout](#)

An instance of the [PlcTypeLayout](#) class which provides the characteristics to use to absolutely align the

members.

members IPlcMemberInfo[]

The IPlcMemberInfo objects to associate with the PlcType.

Returns

PlcType

The PlcType which matches with the characteristics supplied.

Exceptions

ArgumentException

The type is a not supported framework type or a jagged multi-dimensional array type or it can not be used together with the identity specified or identity does not provide the necessary information required to address the type of data.

ArgumentNullException

The type, identity, name, layout or members or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct PlcType objects there the type is either of the type of PlcObject or a derivate of the class.

Of(Type, PlcIdentity, PlcTypeLayout, IEnumerable<IPlcMemberInfo>)

Retrieves the according PlcType which matches the supplied type metadata.

C#

```
public static PlcType Of(Type type, PlcIdentity identity, PlcTypeLayout layout,
IEnumerable<IPlcMemberInfo> members)
```

Parameters

type Type

The framework Type represented by the PlcType.

identity PlcIdentity

The PlcIdentity to which the PlcType does refer.

layout PlcTypeLayout

An instance of the PlcTypeLayout class which provides the characteristics to use to absolutely align the

members.

members IEnumerable<IPlcMemberInfo>

The **IPlcMemberInfo** objects to associate with the **PlcType**.

Returns

PlcType

The **PlcType** which matches with the characteristics supplied.

Exceptions

ArgumentException

The **type** is a not supported framework type or a jagged multi-dimensional array type or it can not be used together with the **identity** specified or **identity** does not provide the necessary information required to address the type of data.

ArgumentNullException

The **type**, **identity**, **layout** or **members** or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct **PlcType** objects there the **type** is either of the type of **PlcObject** or a derivate of the class.

Of(Type, PlcIdentity, PlcTypeLayout, IPlcMemberInfo[])

Retrieves the according **PlcType** which matches the supplied type metadata.

C#

```
public static PlcType Of(Type type, PlcIdentity identity, PlcTypeLayout layout, params
IPlcMemberInfo[] members)
```

Parameters

type Type

The framework **Type** represented by the **PlcType**.

identity PlcIdentity

The **PlcIdentity** to which the **PlcType** does refer.

layout PlcTypeLayout

An instance of the **PlcTypeLayout** class which provides the characteristics to use to absolutely align the **members**.

members IPlcMemberInfo[]

The [IPlcMemberInfo](#) objects to associate with the [PlcType](#).

Returns

[PlcType](#)

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

[ArgumentException](#)

The [type](#) is a not supported framework type or a jagged multi-dimensional array type or it can not be used together with the [identity](#) specified or [identity](#) does not provide the necessary information required to address the type of data.

[ArgumentNullException](#)

The [type](#), [identity](#), [layout](#) or [members](#) or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct [PlcType](#) objects there the [type](#) is either of the type of [PlcObject](#) or a derivate of the class.

Of(Type, PlcTypeLayout, IEnumerable<IPlcMemberInfo>)

Retrieves the according [PlcType](#) which matches the supplied type metadata.

C#

```
public static PlcType Of(Type type, PlcTypeLayout layout, IEnumerable<IPlcMemberInfo> members)
```

Parameters

[type](#) [Type](#)

The framework [Type](#) represented by the [PlcType](#).

[layout](#) [PlcTypeLayout](#)

An instance of the [PlcTypeLayout](#) class which provides the characteristics to use to absolutely align the [members](#).

[members](#) [IEnumerable<IPlcMemberInfo>](#)

The [IPlcMemberInfo](#) objects to associate with the [PlcType](#).

Returns

[PlcType](#)

The **PlcType** which matches with the characteristics supplied.

Exceptions

ArgumentException

The **type** is a not supported framework type or a jagged multi-dimensional array type.

ArgumentNullException

The **type**, **layout** or **members** or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct **PlcType** objects there the **type** is either of the type of **PlcObject** or a derivate of the class.

Of(Type, PlcTypeLayout, IPlcMemberInfo[])

Retrieves the according **PlcType** which matches the supplied type metadata.

C#

```
public static PlcType Of(Type type, PlcTypeLayout layout, params IPlcMemberInfo[] members)
```

Parameters

type Type

The framework **Type** represented by the **PlcType**.

layout PlcTypeLayout

An instance of the **PlcTypeLayout** class which provides the characteristics to use to absolutely align the **members**.

members IPlcMemberInfo[]

The **IPlcMemberInfo** objects to associate with the **PlcType**.

Returns

PlcType

The **PlcType** which matches with the characteristics supplied.

Exceptions

ArgumentException

The **type** is a not supported framework type or a jagged multi-dimensional array type.

ArgumentNullException

The **type**, **layout** or **members** or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct **PlcType** objects there the **type** is either of the type of **PlcObject** or a derivate of the class.

Of<T>()

Retrieves the according **PlcType** which matches the supplied type metadata.

C#

```
public static PlcType Of<T>()
```

Returns

PlcType

The **PlcType** which matches with the characteristics supplied.

Remarks

If **T** is a value type then **PlcType** will be also a value type. If **T** is an array type then **PlcType** will be a array type with the length equals zero. The following **PlcType** can not constructed using this method: * **Date**.

- **Time**.
- **S5Time**.

In general specifying **T** as a **DateTime** will result into a **PlcType** with the **DateTime** and specifying **T** as a **TimeSpan** will result into a **PlcType** with the **TimeOfDay**.

Of<T>(PlcIdentity)

Retrieves the according **PlcType** which matches the supplied type metadata.

C#

```
public static PlcType Of<T>(PlcIdentity identity)
```

Parameters

identity **PlcIdentity**

The **PlcIdentity** to which the **PlcType** does refer.

Returns

PlcType

The **PlcType** which matches with the characteristics supplied.

Exceptions

ArgumentException

The **identity** can not be used together with **T** specified or **identity** does not provide the necessary information required to address the type of data.

ArgumentNullException

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

Remarks

If **T** is a value type then **PlcType** will be also a value type. If **T** is an array type then **PlcType** will be a array type with the length equals zero. The following **PlcType** can not constructed using this method: * **Date**.

- **Time**.
- **S5Time**.

In general specifying **T** as a **DateTime** will result into a **PlcType** with the **DateTime** and specifying **T** as a **TimeSpan** will result into a **PlcType** with the **TimeOfDay**.

Of<T>(PlcIdentity, Int32)

Retrieves the according **PlcType** which matches the supplied type metadata.

C#

```
public static PlcType Of<T>(PlcIdentity identity, int length)
```

Parameters

identity **PlcIdentity**

The **PlcIdentity** to which the **PlcType** does refer.

length **Int32**

The length of the **PlcType** which either specifies the total number of elements in an array type or the total number of characters in a string type.

Returns

PlcType

The **PlcType** which matches with the characteristics supplied.

Exceptions

ArgumentException

The **identity** can not be used together with **T** specified or **identity** does not provide the necessary information required to address the type of data.

ArgumentNullException

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

ArgumentOutOfRangeException

The **length** is out of the bounds defined by **MinLength** and **MaxLength** or **MinLength** and **MaxLength**.

Remarks

This method does always construct array types using the **PlcType**. Therefore it is not possible to use the following framework types as the value for **T**: * **DateTime**.

- **TimeSpan**.
- **PlcObject** or derivate from it.

It is also possible to construct a **PlcType** with the **String** through this method, there **length** does specify the maximum number of characters in the string.

Of<T>(PlcIdentity, PlcName)

Retrieves the according **PlcType** which matches the supplied type metadata.

C#

```
public static PlcType Of<T>(PlcIdentity identity, PlcName name)
```

Parameters

identity **PlcIdentity**

The **PlcIdentity** to which the **PlcType** does refer.

name **PlcName**

The **PlcName** of the **PlcType**.

Returns

PlcType

The **PlcType** which matches with the characteristics supplied.

Exceptions

ArgumentException

The **identity** can not be used together with **T** specified or **identity** does not provide the necessary information required to address the type of data.

ArgumentNullException

The **identity** or **name** is a null reference (Nothing in Visual Basic).

Remarks

If **T** is a value type then **PlcType** will be also a value type. If **T** is an array type then **PlcType** will be a array type with the length equals zero. The following **PlcType** can not constructed using this method: * **Date**.

- **Time**.
- **S5Time**.

In general specifying **T** as a **DateTime** will result into a **PlcType** with the **DateTime** and specifying **T** as a **TimeSpan** will result into a **PlcType** with the **TimeOfDay**.

Of<T>(PlcIdentity, PlcName, Int32)

Retrieves the according **PlcType** which matches the supplied type metadata.

C#

```
public static PlcType Of<T>(PlcIdentity identity, PlcName name, int length)
```

Parameters

identity **PlcIdentity**

The **PlcIdentity** to which the **PlcType** does refer.

name **PlcName**

The **PlcName** of the **PlcType**.

length **Int32**

The length of the **PlcType** which either specifies the total number of elements in an array type or the total number of characters in a string type.

Returns

PlcType

The **PlcType** which matches with the characteristics supplied.

Exceptions

ArgumentException

The **identity** can not be used together with **T** specified or **identity** does not provide the necessary information required to address the type of data.

ArgumentNullException

The **identity** or **name** is a null reference (Nothing in Visual Basic).

ArgumentOutOfRangeException

The `length` is out of the bounds defined by `MinLength` and `MaxLength` or `MinLength` and `MaxLength`.

Remarks

This method does always construct array types using the `PlcType`. Therefore it is not possible to use the following framework types as the value for `T`: * `DateTime`.

- `TimeSpan`.
- `PlcObject` or derivate from it.

It is also possible to construct a `PlcType` with the `String` through this method, there `length` does specify the maximum number of characters in the string.

Of<T>(PlcIdentity, PlcName, PlcTypeLayout, IEnumerable<IPlcMemberInfo>)

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of<T>(PlcIdentity identity, PlcName name, PlcTypeLayout layout,
    IEnumerable<IPlcMemberInfo> members)
```

Parameters

`identity` `PlcIdentity`

The `PlcIdentity` to which the `PlcType` does refer.

`name` `PlcName`

The `PlcName` of the `PlcType`.

`layout` `PlcTypeLayout`

An instance of the `PlcTypeLayout` class which provides the characteristics to use to absolutely align the `members`.

`members` `IEnumerable<IPlcMemberInfo>`

The `IPlcMemberInfo` objects to associate with the `PlcType`.

Returns

`PlcType`

The `PlcType` which matches with the characteristics supplied.

Exceptions

`ArgumentException`

The **identity** can not be used together with **T** specified or **identity** does not provide the necessary information required to address the type of data.

ArgumentNullException

The **identity**, **name**, **layout** or **members** or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct **PlcType** objects there the **T** is either of the type of **PlcObject** or a derivate of the class.

Of<T>(PlcIdentity, PlcName, PlcTypeLayout, IPlcMemberInfo[])

Retrieves the according **PlcType** which matches the supplied type metadata.

C#

```
public static PlcType Of<T>(PlcIdentity identity, PlcName name, PlcTypeLayout layout, params  
IPlcMemberInfo[] members)
```

Parameters

identity **PlcIdentity**

The **PlcIdentity** to which the **PlcType** does refer.

name **PlcName**

The **PlcName** of the **PlcType**.

layout **PlcTypeLayout**

An instance of the **PlcTypeLayout** class which provides the characteristics to use to absolutely align the **members**.

members **IPlcMemberInfo[]**

The **IPlcMemberInfo** objects to associate with the **PlcType**.

Returns

PlcType

The **PlcType** which matches with the characteristics supplied.

Exceptions

ArgumentException

The **identity** can not be used together with **T** specified or **identity** does not provide the necessary information required to address the type of data.

ArgumentNullException

The `identity`, `name`, `layout` or `members` or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct `PlcType` objects there the `T` is either of the type of `PlcObject` or a derivate of the class.

Of<T>(PlcIdentity, PlcTypeLayout, IEnumerable<IPlcMemberInfo>)

Retrieves the according `PlcType` which matches the supplied type metadata.

C#

```
public static PlcType Of<T>(PlcIdentity identity, PlcTypeLayout layout,
    IEnumerable<IPlcMemberInfo> members)
```

Parameters

`identity` `PlcIdentity`

The `PlcIdentity` to which the `PlcType` does refer.

`layout` `PlcTypeLayout`

An instance of the `PlcTypeLayout` class which provides the characteristics to use to absolutely align the `members`.

`members` `IEnumerable<IPlcMemberInfo>`

The `IPlcMemberInfo` objects to associate with the `PlcType`.

Returns

`PlcType`

The `PlcType` which matches with the characteristics supplied.

Exceptions

`ArgumentException`

The `identity` can not be used together with `T` specified or `identity` does not provide the necessary information required to address the type of data.

`ArgumentNullException`

The `identity`, `layout` or `members` or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct [PlcType](#) objects there the **T** is either of the type of [PlcObject](#) or a derivate of the class.

Of<T>(PlcIdentity, PlcTypeLayout, IPlcMemberInfo[])

Retrieves the according [PlcType](#) which matches the supplied type metadata.

C#

```
public static PlcType Of<T>(PlcIdentity identity, PlcTypeLayout layout, params
IPlcMemberInfo[] members)
```

Parameters

identity [PlcIdentity](#)

The [PlcIdentity](#) to which the [PlcType](#) does refer.

layout [PlcTypeLayout](#)

An instance of the [PlcTypeLayout](#) class which provides the characteristics to use to absolutely align the **members**.

members [IPlcMemberInfo\[\]](#)

The [IPlcMemberInfo](#) objects to associate with the [PlcType](#).

Returns

[PlcType](#)

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

[ArgumentException](#)

The **identity** can not be used together with **T** specified or **identity** does not provide the necessary information required to address the type of data.

[ArgumentNullException](#)

The **identity**, **layout** or **members** or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct [PlcType](#) objects there the **T** is either of the type of [PlcObject](#) or a derivate of the class.

Of<T>(PlcTypeLayout, IEnumerable<IPlcMemberInfo>)

Retrieves the according [PlcType](#) which matches the supplied type metadata.

C#

```
public static PlcType Of<T>(PlcTypeLayout layout, IEnumerable<IPlcMemberInfo> members)
```

Parameters

layout [PlcTypeLayout](#)

An instance of the [PlcTypeLayout](#) class which provides the characteristics to use to absolutely align the **members**.

members [IEnumerable<IPlcMemberInfo>](#)

The [IPlcMemberInfo](#) objects to associate with the [PlcType](#).

Returns

[PlcType](#)

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

[ArgumentNullException](#)

The **layout** or **members** or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct [PlcType](#) objects there the **T** is either of the type of [PlcObject](#) or a derivate of the class.

Of<T>(PlcTypeLayout, IPlcMemberInfo[])

Retrieves the according [PlcType](#) which matches the supplied type metadata.

C#

```
public static PlcType Of<T>(PlcTypeLayout layout, params IPlcMemberInfo[] members)
```

Parameters

layout [PlcTypeLayout](#)

An instance of the [PlcTypeLayout](#) class which provides the characteristics to use to absolutely align the **members**.

members [IPlcMemberInfo\[\]](#)

The [IPlcMemberInfo](#) objects to associate with the [PlcType](#).

Returns

PlcType

The [PlcType](#) which matches with the characteristics supplied.

Exceptions

ArgumentNullException

The [Layout](#) or [members](#) or one of its items is a null reference (Nothing in Visual Basic).

Remarks

This method does only construct [PlcType](#) objects there the [T](#) is either of the type of [PlcObject](#) or a derivate of the class.

Relocate(Int32)

Relocates the [PlcType](#) using the specified offset.

C#

```
public PlcType Relocate(int operandNumber)
```

Parameters

operandNumber [Int32](#)

The operand number offset used to adjust the [Address](#).

Returns

PlcType

A new instance of the [PlcType](#) 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 (see [IsAbsolute](#) property).

Relocate(Int32, Int32)

Relocates the [PlcType](#) using the specified offset.

C#

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

Parameters

operandNumber [Int32](#)

The operand number offset used to adjust the [Address](#).

byteNumber [Int32](#)

The byte number offset used to adjust the [Address](#).

Returns

[PlcType](#)

A new instance of the [PlcType](#) 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 (see [IsAbsolute](#) property).

Relocate(Int32, Int32, Int32)

Relocates the [PlcType](#) using the specified offset.

C#

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

Parameters

operandNumber [Int32](#)

The operand number offset used to adjust the [Address](#).

byteNumber [Int32](#)

The byte number offset used to adjust the [Address](#).

bitNumber Int32

The bit number offset used to adjust the [Address](#).

Returns

PlcType

A new instance of the [PlcType](#) 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 (see [IsAbsolute](#) property).

Relocate(PlcAddress)

Relocates the [PlcType](#) using the specified [address](#).

C#

```
public PlcType Relocate(PlcAddress address)
```

Parameters

address PlcAddress

The [PlcAddress](#) to that the type is to be relocated.

Returns

PlcType

A new instance of the [PlcType](#) 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 **PlcRawType** 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 PLC type.

ToString()

Returns a **String** representing the framework type name and the address range of the current **PlcType**.

C#

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

Returns

String

A **String** representing the framework type name and address range of the current **PlcType**.

Table of Contents

Fields	1
MaxLength	1
MinLength	1
Properties	1
Address	1
DefaultValue	2
Description	2
EndAddress	2
FrameworkType	3
HasElementType	3
Identity	3
IsAbsolute	4
IsArray	4
IsClass	4
Layout	5
Length	5
Members	5
Name	5
RelativeType	6
Size	6
SizeInBits	6
TypeCode	7
Methods	7
CreateEntity()	7
Equals(Object)	7
Equals(PlcType)	8
GetElementType(Int32)	8
GetHashCode()	9
GetRawType(PlcTypeCode)	9
GetRawType(Type)	9
GetSize(PlcRawType)	10
GetSize(PlcRawType, Boolean)	10
GetSize(PlcTypeCode)	11
GetSize(PlcTypeCode, Boolean)	11
GetSize(Type)	12
GetSize(Type, Boolean)	12
GetSupportedRawTypes(PlcTypeCode)	13
GetSupportedRawTypes(Type)	13
GetType(PlcTypeCode)	14
GetTypeCode(Type)	14
IsSupported(PlcTypeCode, PlcRawType)	15
IsSupportedArrayType(PlcTypeCode)	15
MakeAbsoluteType(PlcIdentity)	16
MakeArrayType()	16
MakeArrayType(Int32)	17
Of(PlcTypeCode)	17
Of(PlcTypeCode, PlcIdentity)	18
Of(PlcTypeCode, PlcIdentity, Int32)	19
Of(PlcTypeCode, PlcIdentity, PlcName)	20
Of(PlcTypeCode, PlcIdentity, PlcName, Int32)	21
Of(PlcTypeCode, PlcIdentity, PlcName, PlcTypeLayout, IEnumerable<IPlcMemberInfo>)	22