

# PlcDeviceConnection Members

**Namespace:** IPS7Lnk.Advanced

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

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

## Constructors

### PlcDeviceConnection(IPlcDevice)

Initializes a new instance of the [PlcDeviceConnection](#) class using the specified [device](#).

**C#**

```
protected PlcDeviceConnection(IPlcDevice device)
```

#### Parameters

[device](#) [IPlcDevice](#)

The [IPlcDevice](#) associated with the new [PlcDeviceConnection](#).

#### Exceptions

[ArgumentNullException](#)

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

## Fields

### DefaultTimeout

Represents the default timeout value that is used by the [PlcDeviceConnection](#) as initial value of all timeout properties.

**C#**

```
public static readonly int DefaultTimeout
```

#### Field Value

[Int32](#)

## Events

## Closed

Occurs when a connection transitions into the closed state.

**C#**

```
public event EventHandler Closed
```

## Closing

Occurs when a connection transitions into the closing state.

**C#**

```
public event EventHandler Closing
```

## Connected

Occurs when a connection transitions into the connected state.

**C#**

```
public event EventHandler Connected
```

## Connecting

Occurs when a connection transitions into the connecting state.

**C#**

```
public event EventHandler Connecting
```

## Disconnected

Occurs when a connection transitions into the disconnected state.

**C#**

```
public event EventHandler Disconnected
```

## Faulted

Occurs when a connection transitions into the faulted state.

**C#**

```
public event EventHandler Faulted
```

## Opened

Occurs when a connection transitions into the opened state.

**C#**

```
public event EventHandler Opened
```

## Opening

Occurs when a connection transitions into the opening state.

**C#**

```
public event EventHandler Opening
```

## StateChanged

Occurs when a connection transitions into a different state.

**C#**

```
public event EventHandler<PlcDeviceConnectionStateChangedEventArgs> StateChanged
```

## Properties

### BreakDetectionTimeout

Gets or sets the time that is used to detect a connection break.

**C#**

```
public int BreakDetectionTimeout { get; set; }
```

#### Property Value

[Int32](#)

The time in milliseconds to detect a connection break.

#### Exceptions

[ArgumentOutOfRangeException](#)

The value specified is less than zero.

#### Remarks

The default value of this property is [DefaultTimeout](#).

# Channel

Gets the channel object used to communicate with the [Device](#).

## C#

```
protected PlcDeviceConnectionChannel Channel { get; }
```

## Property Value

[PlcDeviceConnectionChannel](#)

The [PlcDeviceConnectionChannel](#) used by the connection to operate on the [Device](#) this connection belongs. The [PlcDeviceConnectionChannel](#) returned can be used by one or more connections.

## Remarks

Only in case there the last connection using a channel gets closed, the channel gets finally released. The shared use of channel objects takes only place in case there the [UsePool](#) property is set to the value true before [Open](#) is called. Any call to [Open](#) before setting the [UsePool](#) property to the value true will not cache a channel instance for pooling. Subsequent calls to [Open](#) after setting the [UsePool](#) property to the value true will share the same channel (in case there the end point and device setup of the connections opening match together).

# ConnectTimeout

Gets or sets the wait time before terminating the attempt to establish a connection.

## C#

```
public int ConnectTimeout { get; set; }
```

## Property Value

[Int32](#)

The time in milliseconds to wait for the connection to connect.

## Exceptions

[ArgumentOutOfRangeException](#)

The value specified is less than zero.

## Remarks

The default value of this property is [DefaultTimeout](#).

## Device

Gets the [IPlcDevice](#) associated with the connection.

### C#

```
public IPlcDevice Device { get; }
```

### Property Value

[IPlcDevice](#)

The [IPlcDevice](#) associated with the connection.

## IsConnected

Gets a value indicating whether the connection has been completely established to the device.

### C#

```
public virtual bool IsConnected { get; }
```

### Property Value

[Boolean](#)

The value true, if the connection has been completely established; otherwise the value false.

## ReceiveTimeout

Gets or sets the wait time before terminating the attempt to receive data.

### C#

```
public int ReceiveTimeout { get; set; }
```

### Property Value

[Int32](#)

The time in milliseconds to wait for the connection to receive.

### Exceptions

[ArgumentOutOfRangeException](#)

The value specified is less than zero.

## Remarks

The default value of this property is [DefaultTimeout](#).

# State

Gets or sets a value that indicates the current state of the connection.

## C#

```
public PlcDeviceConnectionState State { get; protected set; }
```

## Property Value

[PlcDeviceConnectionState](#)

One of the members defined by the [PlcDeviceConnectionState](#) enumeration. Which specifies the current state of the connection.

# Status

Gets the status information provided by the software driver.

## C#

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

## Property Value

[PlcStatus](#)

The latest status information provided by the software driver.

# TransmitTimeout

Gets or sets the wait time before terminating the attempt to transmit data.

## C#

```
public int TransmitTimeout { get; set; }
```

## Property Value

[Int32](#)

The time in milliseconds to wait for the connection to transmit.

## Exceptions

[ArgumentOutOfRangeException](#)

The value specified is less than zero.

### Remarks

The default value of this property is [DefaultTimeout](#).

## UseBreakDetection

Gets or sets a value indicating whether a connection break detection is to be used.

### C#

```
public bool UseBreakDetection { get; set; }
```

### Property Value

[Boolean](#)

The value true, if a connection break detection is to be used; otherwise the value false.

### Remarks

The default value of this property is false.

## UsePool

Gets or sets a value indicating whether [PlcDeviceConnection](#) instances shall use a mechanism to reuse existing channels with the same connection setup.

### C#

```
public static bool UsePool { get; set; }
```

### Property Value

[Boolean](#)

The value true if existing channels shall be reused with the same connection setup; otherwise the value false.

## Methods

### ClearAllPools()

Empties the connection pool.

### C#

```
public static void ClearAllPools()
```

## Remarks

[ClearAllPools](#) clears the whole connection pool. If there are connections in use at the time of the call, they are closed.

# ClearPool(PlcDeviceConnection)

Empties the connection pool associated with the specified [connection](#).

## C#

```
public static void ClearPool(PlcDeviceConnection connection)
```

## Parameters

[connection](#) [PlcDeviceConnection](#)

The [PlcDeviceConnection](#) to be cleared from the pool.

## Exceptions

[ArgumentNullException](#)

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

## Remarks

[ClearPool\(PlcDeviceConnection\)](#) clears the connection pool that is associated with the connection. If additional connections associated with connection are in use at the time of the call, they are closed.

# Close()

Causes a connection to transition from its current state into the closed state while an established connection to a device will be closed.

## C#

```
public void Close()
```

## Exceptions

[InvalidOperationException](#)

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

[ObjectDisposedException](#)



The connection has been disposed of.

## CloseCore()

Closes an established connection to a device.

**C#**

```
protected virtual void CloseCore()
```

## Connect()

Causes a connection to transition from the opened state into the connected state while it does fully establish a connection to a device.

**C#**

```
public void Connect()
```

### Exceptions

#### [InvalidOperationException](#)

The connection is not in [Opened](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

#### [ObjectDisposedException](#)

The connection has been disposed of.

### Remarks

In case there is already a connection fully established (see [IsConnected](#)) [Connect](#) does not perform any connect attempt.

## ConnectCore()

Fully establishes a connection to a device.

**C#**

```
protected virtual void ConnectCore()
```

## CreateChannel()

When implemented in a derived class, creates a new immutable channel which is used as the low level access layer of the connection.

**C#**

```
protected abstract PlcDeviceConnectionChannel CreateChannel()
```

## Returns

[PlcDeviceConnectionChannel](#)

A new [PlcDeviceConnectionChannel](#) configured with the setup of this [PlcDeviceConnection](#) and its [Device](#).

# CreateNode(PlcObject)

Creates a new [PlcDataNode](#) which refers to a [PlcObject](#).

## C#

```
protected virtual PlcDataNode CreateNode(PlcObject instance)
```

## Parameters

**instance** [PlcObject](#)

The [PlcObject](#) for that the node is to be created.

## Returns

[PlcDataNode](#)

A new [PlcDataNode](#) representing the access layer for the [PlcObject](#) specified by **instance**.

## Exceptions

[ArgumentNullException](#)

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

# CreateNode(PlcType)

Creates a new [PlcDataNode](#) which refers to a [PlcType](#).

## C#

```
protected virtual PlcDataNode CreateNode(PlcType type)
```

## Parameters

**type** [PlcType](#)

The [PlcType](#) for that the node is to be created.

## Returns

## PlcDataNode

A new [PlcDataNode](#) representing the access layer for the [PlcType](#) specified by [type](#).

### Exceptions

#### [ArgumentNullException](#)

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

## CreateNode(PlcType, IPlcValue)

Creates a new [PlcDataNode](#) which refers to a [PlcType](#) and [IPlcValue](#).

### C#

```
protected virtual PlcDataNode CreateNode(PlcType type, IPlcValue value)
```

### Parameters

[type](#) [PlcType](#)

The [PlcType](#) for that the node is to be created.

[value](#) [IPlcValue](#)

The [IPlcValue](#) used by the node to represent the [type](#) specific data.

### Returns

[PlcDataNode](#)

A new [PlcDataNode](#) representing the access layer for the [PlcType](#) specified by [type](#) and [IPlcValue](#) defined by [value](#).

## DenyIfFaulted()

Verifies whether the connection is in faulted state.

### C#

```
protected void DenyIfFaulted()
```

### Exceptions

#### [InvalidOperationException](#)

The connection is in faulted state.

## DenyIfIsDisposed()

Verifies whether the connection has been disposed of.

**C#**

```
protected void DenyIfIsDisposed()
```

### Exceptions

[ObjectDisposedException](#)

The connection has been disposed of.

## DenyIfNotReady()

Verifies whether the connection is in opened, connected or disconnected state.

**C#**

```
protected void DenyIfNotReady()
```

## Dispose()

Releases all resources used by the [PlcDeviceConnection](#).

**C#**

```
public void Dispose()
```

## Dispose(Boolean)

Releases the unmanaged resources used by the [PlcDeviceConnection](#) and optionally releases the managed resources.

**C#**

```
protected virtual void Dispose(bool disposing)
```

### Parameters

**disposing** [Boolean](#)

The value true to release both managed and unmanaged resources; otherwise the value false to release only unmanaged resources.

## Finalize()

Finalizes an instance of the [PlcDeviceConnection](#) class.

C#

```
protected void Finalize()
```

## GetStatus(PlcType)

Retrieves the most recent [PlcStatus](#) resulted by the most recent read/write action performed using the specified [type](#).

C#

```
public PlcStatus GetStatus(PlcType type)
```

### Parameters

[type](#) [PlcType](#)

The [PlcType](#) its most recent [PlcStatus](#) is to be retrieved.

### Returns

[PlcStatus](#)

The [PlcStatus](#) of the most recent action performed using the specified [type](#).

### Exceptions

[ArgumentNullException](#)

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

## OnClosed(EventArgs)

Raises the [Closed](#) event of the [PlcDeviceConnection](#).

C#

```
protected virtual void OnClosed(EventArgs e)
```

### Parameters

[e](#) [EventArgs](#)

The event data.

## OnClosing(EventArgs)

Raises the [Closing](#) event of the [PlcDeviceConnection](#).

C#

```
protected virtual void OnClosing(EventArgs e)
```

### Parameters

**e** [EventArgs](#)

The event data.

## OnConnected(EventArgs)

Raises the [Connected](#) event of the [PlcDeviceConnection](#).

### C#

```
protected virtual void OnConnected(EventArgs e)
```

### Parameters

**e** [EventArgs](#)

The event data.

## OnConnecting(EventArgs)

Raises the [Connecting](#) event of the [PlcDeviceConnection](#).

### C#

```
protected virtual void OnConnecting(EventArgs e)
```

### Parameters

**e** [EventArgs](#)

The event data.

## OnDisconnected(EventArgs)

Raises the [Disconnected](#) event of the [PlcDeviceConnection](#).

### C#

```
protected virtual void OnDisconnected(EventArgs e)
```

### Parameters

**e** [EventArgs](#)

The event data.

## OnFaulted(EventArgs)

Raises the [Faulted](#) event of the [PlcDeviceConnection](#).

**C#**

```
protected virtual void OnFaulted(EventArgs e)
```

### Parameters

**e** [EventArgs](#)

The event data.

## OnOpened(EventArgs)

Raises the [Opened](#) event of the [PlcDeviceConnection](#).

**C#**

```
protected virtual void OnOpened(EventArgs e)
```

### Parameters

**e** [EventArgs](#)

The event data.

## OnOpening(EventArgs)

Raises the [Opening](#) event of the [PlcDeviceConnection](#).

**C#**

```
protected virtual void OnOpening(EventArgs e)
```

### Parameters

**e** [EventArgs](#)

The event data.

## OnStateChanged(PlcDeviceConnectionStateChangedEventArgs)

Raises the [StateChanged](#) event of the [PlcDeviceConnection](#).

**C#**

```
protected virtual void OnStateChanged(PlcDeviceConnectionStateChangedEventArgs e)
```

## Parameters

e [PlcDeviceConnectionStateChangedEventArgs](#)

The event data.

## Open()

Causes a connection to transition from the created or closed state into the opened state while it does establish a connection to a device.

### C#

```
public void Open()
```

## Exceptions

[InvalidOperationException](#)

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

[ObjectDisposedException](#)

The connection has been disposed of.

## OpenCore()

Establishes a connection to a device.

### C#

```
protected virtual void OpenCore()
```

## Read(IEnumerable<PlcIdentity>)

Reads the values associated with the [identities](#) specified.

### C#

```
public IDictionary<PlcIdentity, object> Read(IEnumerable<PlcIdentity> identities)
```

## Parameters

[identities](#) [IEnumerable<PlcIdentity>](#)

The [PlcIdentity](#)'s that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[IDictionary<PlcIdentity, Object>](#)



A dictionary there the [PlcIdentity](#) defines the key, while the red value is associated with the key as value.

## Exceptions

### [ArgumentException](#)

It is not possible to read data using a relative type (see [IsAbsolute](#)).

### [ArgumentNullException](#)

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

### [InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

### [ObjectDisposedException](#)

The connection has been disposed of.

## Read(IEnumerable<PlcType>)

Reads the values associated with the [types](#) specified.

### C#

```
public IDictionary<PlcType, object> Read(IEnumerable<PlcType> types)
```

## Parameters

[types](#) [IEnumerable<PlcType>](#)

The [PlcType](#)'s its values are to be read.

## Returns

[IDictionary<PlcType, Object>](#)

A dictionary there the [PlcType](#) defines the key, while the red value is associated with the key as value.

## Exceptions

### [ArgumentException](#)

It is not possible to read data using a relative type (see [IsAbsolute](#)).

### [ArgumentNullException](#)

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

### [InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the

connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## Read(IEnumerable<String>)

Reads the values associated with the **identities** specified.

### C#

```
public IDictionary<string, object> Read(IEnumerable<string> identities)
```

### Parameters

**identities** [IEnumerable<String>](#)

The [PlcIdentity](#)'s that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

### Returns

[IDictionary<String, Object>](#)

A dictionary where the [PlcIdentity](#) defines the key, while the red value is associated with the key as value.

### Exceptions

#### ArgumentException

One of the **identities** specified is an empty string or it is not possible to read data using a relative type (see [IsAbsolute](#)).

#### ArgumentNullException

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

#### InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## Read(PlcIdentity[])

Reads the values associated with the **identities** specified.

### C#

```
public IDictionary<PlcIdentity, object> Read(params PlcIdentity[] identities)
```

## Parameters

**identities** [PlcIdentity\[\]](#)

The [PlcIdentity](#)'s that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[IDictionary<PlcIdentity, Object>](#)

A dictionary where the [PlcIdentity](#) defines the key, while the red value is associated with the key as value.

## Exceptions

[ArgumentException](#)

It is not possible to read data using a relative type (see [IsAbsolute](#)).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# Read(PlcType[])

Reads the values associated with the **types** specified.

## C#

```
public IDictionary<PlcType, object> Read(params PlcType[] types)
```

## Parameters

**types** [PlcType\[\]](#)

The [PlcType](#)'s its values are to be read.

## Returns

[IDictionary<PlcType, Object>](#)

A dictionary where the [PlcType](#) defines the key, while the red value is associated with the key as value.

## Exceptions

### ArgumentException

It is not possible to read data using a relative type (see [IsAbsolute](#)).

### ArgumentNullException

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

### InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## Read(String[])

Reads the values associated with the `identities` specified.

### C#

```
public IDictionary<string, object> Read(params string[] identities)
```

## Parameters

`identities` [String](#)[]

The [PlcIdentity](#)'s that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[IDictionary](#)<[String](#), [Object](#)>

A dictionary where the [PlcIdentity](#) defines the key, while the red value is associated with the key as value.

## Exceptions

### ArgumentException

One of the `identities` specified is an empty string or it is not possible to read data using a relative type (see [IsAbsolute](#)).

### ArgumentNullException

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

### InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# ReadBoolean(PlcBooleanArrayType)

Reads a [Boolean](#) array (in PLC an ARRAY OF BOOL) using the [type](#) specified.

## C#

```
public bool[] ReadBoolean(PlcBooleanArrayType type)
```

## Parameters

[type](#) [PlcBooleanArrayType](#)

The [PlcBooleanArrayType](#) which addresses the data area to read.

## Returns

[Boolean\[\]](#)

The array stored at the data area described by the [type](#) specified.

## Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadBoolean(PlcBooleanType)

Reads a [Boolean](#) (in PLC a BOOL) using the [type](#) specified.

## C#

```
public bool ReadBoolean(PlcBooleanType type)
```

## Parameters

[type](#) [PlcBooleanType](#)

The [PlcBooleanType](#) which addresses the data area to read.

## Returns

### Boolean

The value stored at the data area described by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# ReadBoolean(PlcIdentity)

Reads a `Boolean` (in PLC a `BOOL`) which is associated with the `identity` specified.

## C#

```
public bool ReadBoolean(PlcIdentity identity)
```

## Parameters

### `identity` `PlcIdentity`

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to read. Or a `PlcName` representing the symbolic name of the `PlcType` (to read) associated with the `Device` of this connection.

## Returns

### Boolean

The value stored at the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` does not refer to a `PlcType` that addresses data of the raw type `Bit`.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## ReadBoolean(PlcIdentity, Int32)

Reads a [Boolean](#) array (in PLC an ARRAY OF BOOL) which is associated with the [identity](#) specified.

**C#**

```
public bool[] ReadBoolean(PlcIdentity identity, int count)
```

### Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

[count](#) [Int32](#)

The number of items to read.

### Returns

[Boolean](#)[]

The array stored at the data area identified by the [identity](#) specified.

### Exceptions

[ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [Bit](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadBoolean(String)

Reads a **Boolean** (in PLC a BOOL) which is associated with the **identity** specified.

**C#**

```
public bool ReadBoolean(string identity)
```

## Parameters

**identity** **String**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to read. Or a **PlcName** representing the symbolic name of the **PlcType** (to read) associated with the **Device** of this connection.

## Returns

**Boolean**

The value stored at the data area identified by the **identity** specified.

## Exceptions

**ArgumentException**

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Bit**.

**ArgumentNullException**

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

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# ReadBoolean(String, Int32)

Reads a **Boolean** array (in PLC an ARRAY OF BOOL) which is associated with the **identity** specified.

**C#**

```
public bool[] ReadBoolean(string identity, int count)
```

## Parameters

**identity** **String**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to read. Or a **PlcName** representing the symbolic name of the **PlcType** (to read) associated with the **Device** of this connection.



count Int32

The number of items to read.

## Returns

Boolean[]

The array stored at the data area identified by the `identity` specified.

## Exceptions

ArgumentException

The `identity` is an empty string or does not refer to a `PlcType` that addresses data of the raw type `Bit`.

ArgumentNullException

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

ArgumentOutOfRangeException

The `count` is outside of the bounds defined by `MinLength` and `MaxLength`.

InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

ObjectDisposedException

The connection has been disposed of.

# ReadBooleanCore(PlcType)

Reads the data area described by the `type` specified.

C#

```
protected virtual bool[] ReadBooleanCore(PlcType type)
```

## Parameters

type PlcType

The `PlcBooleanType` or `PlcBooleanArrayType` to read.

## Returns

Boolean[]

An array of the `Boolean` data read.

# ReadByte(PlcByteArrayType)

Reads a [Byte](#) array (in PLC an ARRAY OF BYTE) using the [type](#) specified.

## C#

```
public byte[] ReadByte(PlcByteArrayType type)
```

### Parameters

[type](#) [PlcByteArrayType](#)

The [PlcByteArrayType](#) which addresses the data area to read.

### Returns

[Byte](#)[]

The array stored at the data area described by the [type](#) specified.

### Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadByte(PlcByteType)

Reads a [Byte](#) (in PLC a BYTE) using the [type](#) specified.

## C#

```
public byte ReadByte(PlcByteType type)
```

### Parameters

[type](#) [PlcByteType](#)

The [PlcByteType](#) which addresses the data area to read.

### Returns

[Byte](#)

The value stored at the data area described by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# ReadByte(PlcIdentity)

Reads a `Byte` (in PLC a `BYTE`) which is associated with the `identity` specified.

## C#

```
public byte ReadByte(PlcIdentity identity)
```

## Parameters

`identity` `PlcIdentity`

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to read. Or a `PlcName` representing the symbolic name of the `PlcType` (to read) associated with the `Device` of this connection.

## Returns

`Byte`

The value stored at the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` does not refer to a `PlcType` that addresses data of the raw type `Byte`.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# ReadByte(PlcIdentity, Int32)

Reads a [Byte](#) array (in PLC an ARRAY OF BYTE) which is associated with the [identity](#) specified.

## C#

```
public byte[] ReadByte(PlcIdentity identity, int count)
```

## Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

[count](#) [Int32](#)

The number of items to read.

## Returns

[Byte](#)[]

The array stored at the data area identified by the [identity](#) specified.

## Exceptions

[ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [Byte](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadByte(String)

Reads a [Byte](#) (in PLC a BYTE) which is associated with the [identity](#) specified.

## C#

```
public byte ReadByte(string identity)
```

### Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

### Returns

[Byte](#)

The value stored at the data area identified by the [identity](#) specified.

### Exceptions

[ArgumentException](#)

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [Byte](#).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadByte(String, Int32)

Reads a [Byte](#) array (in PLC an ARRAY OF BYTE) which is associated with the [identity](#) specified.

## C#

```
public byte[] ReadByte(string identity, int count)
```

### Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

count Int32

The number of items to read.

## Returns

Byte[]

The array stored at the data area identified by the `identity` specified.

## Exceptions

ArgumentException

The `identity` is an empty string or does not refer to a `PlcType` that addresses data of the raw type `Byte`.

ArgumentNullException

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

ArgumentOutOfRangeException

The `count` is outside of the bounds defined by `MinLength` and `MaxLength`.

InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

ObjectDisposedException

The connection has been disposed of.

# ReadByteCore(PlcType)

Reads the data area described by the `type` specified.

C#

```
protected virtual byte[] ReadByteCore(PlcType type)
```

## Parameters

type PlcType

The `PlcByteType` or `PlcByteArrayType` to read.

## Returns

Byte[]

An array of the `Byte` data read.

# ReadChar(PlcCharArrayType)

Reads a [Char](#) array (in PLC an ARRAY OF CHAR) using the [type](#) specified.

**C#**

```
public char[] ReadChar(PlcCharArrayType type)
```

## Parameters

[type](#) [PlcCharArrayType](#)

The [PlcCharArrayType](#) which addresses the data area to read.

## Returns

[Char\[\]](#)

The array stored at the data area described by the [type](#) specified.

## Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadChar(PlcCharType)

Reads a [Char](#) (in PLC a CHAR) using the [type](#) specified.

**C#**

```
public char ReadChar(PlcCharType type)
```

## Parameters

[type](#) [PlcCharType](#)

The [PlcCharType](#) which addresses the data area to read.

## Returns

[Char](#)

The value stored at the data area described by the **type** specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# ReadChar(PlcIdentity)

Reads a **Char** (in PLC a CHAR) which is associated with the **identity** specified.

## C#

```
public char ReadChar(PlcIdentity identity)
```

## Parameters

### **identity** PlcIdentity

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to read. Or a **PlcName** representing the symbolic name of the **PlcType** (to read) associated with the **Device** of this connection.

## Returns

### Char

The value stored at the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** does not refer to a **PlcType** that addresses data of the raw type **Byte**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.



## ObjectDisposedException

The connection has been disposed of.

# ReadChar(PlcIdentity, Int32)

Reads a [Char](#) array (in PLC an ARRAY OF CHAR) which is associated with the [identity](#) specified.

## C#

```
public char[] ReadChar(PlcIdentity identity, int count)
```

## Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

[count](#) [Int32](#)

The number of items to read.

## Returns

[Char\[\]](#)

The array stored at the data area identified by the [identity](#) specified.

## Exceptions

[ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [Byte](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadChar(String)

Reads a [Char](#) (in PLC a CHAR) which is associated with the [identity](#) specified.

**C#**

```
public char ReadChar(string identity)
```

## Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[Char](#)

The value stored at the data area identified by the [identity](#) specified.

## Exceptions

[ArgumentException](#)

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [Byte](#).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadChar(String, Int32)

Reads a [Char](#) array (in PLC an ARRAY OF CHAR) which is associated with the [identity](#) specified.

**C#**

```
public char[] ReadChar(string identity, int count)
```

## Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

`count` `Int32`

The number of items to read.

## Returns

`Char[]`

The array stored at the data area identified by the `identity` specified.

## Exceptions

`ArgumentException`

The `identity` is an empty string or does not refer to a `PlcType` that addresses data of the raw type `Byte`.

`ArgumentNullException`

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

`ArgumentOutOfRangeException`

The `count` is outside of the bounds defined by `MinLength` and `MaxLength`.

`InvalidOperationException`

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

`ObjectDisposedException`

The connection has been disposed of.

# ReadCharCore(PlcType)

Reads the data area described by the `type` specified.

**C#**

```
protected virtual char[] ReadCharCore(PlcType type)
```

## Parameters

`type` `PlcType`

The `PlcCharType` or `PlcCharArrayType` to read.

## Returns

`Char[]`

An array of the `Char` data read.

## ReadCore(IEnumerable<PlcType>)

Reads the data areas described by the **types** specified.

**C#**

```
protected virtual IDictionary<PlcType, object> ReadCore(IEnumerable<PlcType> types)
```

### Parameters

**types** IEnumerable<PlcType>

The PlcType's to read.

### Returns

IDictionary<PlcType, Object>

A dictionary where the PlcType defines the key, while the red value is associated with the key as the value.

## ReadCore(PlcType[])

Reads the data areas described by the **types** specified.

**C#**

```
protected virtual IDictionary<PlcType, object> ReadCore(params PlcType[] types)
```

### Parameters

**types** PlcType[]

The PlcType's to read.

### Returns

IDictionary<PlcType, Object>

A dictionary where the PlcType defines the key, while the red value is associated with the key as the value.

## ReadDate(PlcDateType)

Reads a DateTime (in PLC a DATE) using the **type** specified.

**C#**

```
public DateTime ReadDate(PlcDateType type)
```

### Parameters

**type** PlcDateType

The [PlcDateType](#) which addresses the data area to read.

## Returns

[DateTime](#)

The value stored at the data area described by the [type](#) specified.

## Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadDate(PlcIdentity)

Reads a [DateTime](#) (in PLC a DATE) which is associated with the [identity](#) specified.

**C#**

```
public DateTime ReadDate(PlcIdentity identity)
```

## Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[DateTime](#)

The value stored at the data area identified by the [identity](#) specified.

## Exceptions

[ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [Word](#).

[ArgumentNullException](#)

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

#### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## ReadDate(String)

Reads a `DateTime` (in PLC a DATE) which is associated with the `identity` specified.

### C#

```
public DateTime ReadDate(string identity)
```

### Parameters

#### `identity` String

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to read. Or a `PlcName` representing the symbolic name of the `PlcType` (to read) associated with the `Device` of this connection.

### Returns

#### `DateTime`

The value stored at the data area identified by the `identity` specified.

### Exceptions

#### ArgumentException

The `identity` is an empty string or does not refer to a `PlcType` that addresses data of the raw type `Word`.

#### ArgumentNullException

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

#### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

# ReadDateCore(PlcType)

Reads the data area described by the `type` specified.

**C#**

```
protected virtual DateTime ReadDateCore(PlcType type)
```

## Parameters

`type PlcType`

The `PlcDateType` to read.

## Returns

`DateTime`

The `DateTime` data read.

# ReadDateTime(PlcDateTimeType)

Reads a `DateTime` (in PLC a DATE\_AND\_TIME) using the `type` specified.

**C#**

```
public DateTime ReadDateTime(PlcDateTimeType type)
```

## Parameters

`type PlcDateTimeType`

The `PlcDateTimeType` which addresses the data area to read.

## Returns

`DateTime`

The value stored at the data area described by the `type` specified.

## Exceptions

`ArgumentNullException`

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

`InvalidOperationException`

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

`ObjectDisposedException`

The connection has been disposed of.

## ReadDateTime(PlcIdentity)

Reads a [DateTime](#) (in PLC a DATE\_AND\_TIME) which is associated with the [identity](#) specified.

**C#**

```
public DateTime ReadDateTime(PlcIdentity identity)
```

### Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

### Returns

[DateTime](#)

The value stored at the data area identified by the [identity](#) specified.

### Exceptions

[ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [Byte](#).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## ReadDateTime(String)

Reads a [DateTime](#) (in PLC a DATE\_AND\_TIME) which is associated with the [identity](#) specified.

**C#**

```
public DateTime ReadDateTime(string identity)
```

### Parameters



## identity String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to read. Or a **PlcName** representing the symbolic name of the **PlcType** (to read) associated with the **Device** of this connection.

## Returns

### DateTime

The value stored at the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Byte**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# ReadDateTimeCore(PlcType)

Reads the data area described by the **type** specified.

## C#

```
protected virtual DateTime ReadDateTimeCore(PlcType type)
```

## Parameters

### type PlcType

The **PlcDateTimeType** to read.

## Returns

### DateTime

The **DateTime** data read.

# ReadDouble(PlcDoubleArrayType)

Reads a [Double](#) array (in PLC an ARRAY OF DOUBLE) using the [type](#) specified.

**C#**

```
public double[] ReadDouble(PlcDoubleArrayType type)
```

## Parameters

[type](#) [PlcDoubleArrayType](#)

The [PlcDoubleArrayType](#) which addresses the data area to read.

## Returns

[Double\[\]](#)

The array stored at the data area described by the [type](#) specified.

## Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadDouble(PlcDoubleType)

Reads a [Double](#) (in PLC a DOUBLE) using the [type](#) specified.

**C#**

```
public double ReadDouble(PlcDoubleType type)
```

## Parameters

[type](#) [PlcDoubleType](#)

The [PlcDoubleType](#) which addresses the data area to read.

## Returns

[Double](#)

The value stored at the data area described by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# ReadDouble(PlcIdentity)

Reads a `Double` (in PLC a `DOUBLE`) which is associated with the `identity` specified.

## C#

```
public double ReadDouble(PlcIdentity identity)
```

## Parameters

### `identity` PlcIdentity

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to read. Or a `PlcName` representing the symbolic name of the `PlcType` (to read) associated with the `Device` of this connection.

## Returns

### Double

The value stored at the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` does not refer to a `PlcType` that addresses data of the raw type `DWord`.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# ReadDouble(PlcIdentity, Int32)

Reads a [Double](#) array (in PLC an ARRAY OF DOUBLE) which is associated with the [identity](#) specified.

## C#

```
public double[] ReadDouble(PlcIdentity identity, int count)
```

## Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

[count](#) [Int32](#)

The number of items to read.

## Returns

[Double\[\]](#)

The array stored at the data area identified by the [identity](#) specified.

## Exceptions

[ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadDouble(String)

Reads a [Double](#) (in PLC a DOUBLE) which is associated with the [identity](#) specified.

## C#

```
public double ReadDouble(string identity)
```

### Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

### Returns

[Double](#)

The value stored at the data area identified by the [identity](#) specified.

### Exceptions

[ArgumentException](#)

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadDouble(String, Int32)

Reads a [Double](#) array (in PLC an ARRAY OF DOUBLE) which is associated with the [identity](#) specified.

## C#

```
public double[] ReadDouble(string identity, int count)
```

### Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

count Int32

The number of items to read.

## Returns

Double[]

The array stored at the data area identified by the `identity` specified.

## Exceptions

ArgumentException

The `identity` is an empty string or does not refer to a `PlcType` that addresses data of the raw type `DWord`.

ArgumentNullException

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

ArgumentOutOfRangeException

The `count` is outside of the bounds defined by `MinLength` and `MaxLength`.

InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

ObjectDisposedException

The connection has been disposed of.

# ReadDoubleCore(PlcType)

Reads the data area described by the `type` specified.

C#

```
protected virtual double[] ReadDoubleCore(PlcType type)
```

## Parameters

type PlcType

The `PlcDoubleType` or `PlcDoubleArrayType` to read.

## Returns

Double[]

An array of the `Double` data read.

# ReadEntities(IEnumerable<IPlcEntity>)

Reads the **entities** specified.

**C#**

```
public IEnumerable<PlcEntityReadResult> ReadEntities(IEnumerable<IPlcEntity> entities)
```

## Parameters

**entities** [IEnumerable<IPlcEntity>](#)

The [IPlcEntity](#)'s to read.

## Returns

[IEnumerable<PlcEntityReadResult>](#)

A sequence of [PlcEntityReadResult](#) instances containing the [IPlcEntity](#) and its resulting data after a read operation has been performed for the entity.

## Exceptions

[ArgumentException](#)

It is not possible to read data using a relative type (see [IsAbsolute](#)).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadEntities(IPlcEntity[])

Reads the **entities** specified.

**C#**

```
public IEnumerable<PlcEntityReadResult> ReadEntities(params IPlcEntity[] entities)
```

## Parameters

**entities** [IPlcEntity\[\]](#)

The [IPlcEntity](#)'s to read.

## Returns

[IEnumerable<PlcEntityReadResult>](#)

A sequence of [PlcEntityReadResult](#) instances containing the [IPlcEntity](#) and its resulting data after a read operation has been performed for the entity.

## Exceptions

[ArgumentException](#)

It is not possible to read data using a relative type (see [IsAbsolute](#)).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadInt16(PlcIdentity)

Reads a [Int16](#) (in PLC a INT) which is associated with the [identity](#) specified.

## C#

```
public short ReadInt16(PlcIdentity identity)
```

## Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[Int16](#)

The value stored at the data area identified by the [identity](#) specified.

## Exceptions

[ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [Word](#).



## ArgumentNullException

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

## InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# ReadInt16(PlcIdentity, Int32)

Reads a **Int16** array (in PLC an ARRAY OF INT) which is associated with the **identity** specified.

## C#

```
public short[] ReadInt16(PlcIdentity identity, int count)
```

## Parameters

**identity** **PlcIdentity**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to read. Or a **PlcName** representing the symbolic name of the **PlcType** (to read) associated with the **Device** of this connection.

**count** **Int32**

The number of items to read.

## Returns

**Int16[]**

The array stored at the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** does not refer to a **PlcType** that addresses data of the raw type **Word**.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The **count** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# ReadInt16(PlcInt16ArrayType)

Reads a [Int16](#) array (in PLC an ARRAY OF INT) using the [type](#) specified.

## C#

```
public short[] ReadInt16(PlcInt16ArrayType type)
```

## Parameters

[type](#) [PlcInt16ArrayType](#)

The [PlcInt16ArrayType](#) which addresses the data area to read.

## Returns

[Int16\[\]](#)

The array stored at the data area described by the [type](#) specified.

## Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadInt16(PlcInt16Type)

Reads a [Int16](#) (in PLC a INT) using the [type](#) specified.

## C#

```
public short ReadInt16(PlcInt16Type type)
```

## Parameters

[type](#) [PlcInt16Type](#)

The [PlcInt16Type](#) which addresses the data area to read.

## Returns

### Int16

The value stored at the data area described by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# ReadInt16(String)

Reads a `Int16` (in PLC a `INT`) which is associated with the `identity` specified.

## C#

```
public short ReadInt16(string identity)
```

## Parameters

### identity String

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to read. Or a `PlcName` representing the symbolic name of the `PlcType` (to read) associated with the `Device` of this connection.

## Returns

### Int16

The value stored at the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` is an empty string or does not refer to a `PlcType` that addresses data of the raw type `Word`.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

#### [ObjectDisposedException](#)

The connection has been disposed of.

## ReadInt16(String, Int32)

Reads a [Int16](#) array (in PLC an ARRAY OF INT) which is associated with the [identity](#) specified.

### C#

```
public short[] ReadInt16(string identity, int count)
```

### Parameters

#### [identity](#) String

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

#### [count](#) Int32

The number of items to read.

### Returns

#### [Int16\[\]](#)

The array stored at the data area identified by the [identity](#) specified.

### Exceptions

#### [ArgumentException](#)

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [Word](#).

#### [ArgumentNullException](#)

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

#### [ArgumentOutOfRangeException](#)

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

#### [InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

#### [ObjectDisposedException](#)

The connection has been disposed of.

# ReadInt16Core(PlcType)

Reads the data area described by the **type** specified.

## C#

```
protected virtual short[] ReadInt16Core(PlcType type)
```

## Parameters

**type** PlcType

The PlcInt16Type or PlcInt16ArrayType to read.

## Returns

Int16[]

An array of the Int16 data read.

# ReadInt32(PlcIdentity)

Reads a Int32 (in PLC a DINT) which is associated with the **identity** specified.

## C#

```
public int ReadInt32(PlcIdentity identity)
```

## Parameters

**identity** PlcIdentity

The PlcIdentity that can be either a PlcAddress which addresses the data area to read. Or a PlcName representing the symbolic name of the PlcType (to read) associated with the Device of this connection.

## Returns

Int32

The value stored at the data area identified by the **identity** specified.

## Exceptions

ArgumentException

The **identity** does not refer to a PlcType that addresses data of the raw type DWord.

ArgumentNullException

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

InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## ReadInt32(PlcIdentity, Int32)

Reads a [Int32](#) array (in PLC an ARRAY OF DINT) which is associated with the [identity](#) specified.

**C#**

```
public int[] ReadInt32(PlcIdentity identity, int count)
```

### Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

[count](#) [Int32](#)

The number of items to read.

### Returns

[Int32\[\]](#)

The array stored at the data area identified by the [identity](#) specified.

### Exceptions

[ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadInt32(PlcInt32ArrayType)

Reads a [Int32](#) array (in PLC an ARRAY OF DINT) using the [type](#) specified.

## C#

```
public int[] ReadInt32(PlcInt32ArrayType type)
```

### Parameters

[type](#) [PlcInt32ArrayType](#)

The [PlcInt32ArrayType](#) which addresses the data area to read.

### Returns

[Int32\[\]](#)

The array stored at the data area described by the [type](#) specified.

### Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadInt32(PlcInt32Type)

Reads a [Int32](#) (in PLC a DINT) using the [type](#) specified.

## C#

```
public int ReadInt32(PlcInt32Type type)
```

### Parameters

[type](#) [PlcInt32Type](#)

The [PlcInt32Type](#) which addresses the data area to read.

### Returns

[Int32](#)

The value stored at the data area described by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# ReadInt32(String)

Reads a `Int32` (in PLC a DINT) which is associated with the `identity` specified.

## C#

```
public int ReadInt32(string identity)
```

## Parameters

### `identity` String

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to read. Or a `PlcName` representing the symbolic name of the `PlcType` (to read) associated with the `Device` of this connection.

## Returns

### Int32

The value stored at the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` is an empty string or does not refer to a `PlcType` that addresses data of the raw type `DWord`.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.



## ObjectDisposedException

The connection has been disposed of.

# ReadInt32(String, Int32)

Reads a [Int32](#) array (in PLC an ARRAY OF DINT) which is associated with the [identity](#) specified.

## C#

```
public int[] ReadInt32(string identity, int count)
```

## Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

[count](#) [Int32](#)

The number of items to read.

## Returns

[Int32\[\]](#)

The array stored at the data area identified by the [identity](#) specified.

## Exceptions

[ArgumentException](#)

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadInt32Core(PlcType)

Reads the data area described by the **type** specified.

**C#**

```
protected virtual int[] ReadInt32Core(PlcType type)
```

## Parameters

**type** [PlcType](#)

The [PlcInt32Type](#) or [PlcInt32ArrayType](#) to read.

## Returns

[Int32\[\]](#)

An array of the [Int32](#) data read.

# ReadInt64(PlcIdentity)

Reads a [Int64](#) (in PLC a DINT) which is associated with the **identity** specified.

**C#**

```
public long ReadInt64(PlcIdentity identity)
```

## Parameters

**identity** [PlcIdentity](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[Int64](#)

The value stored at the data area identified by the **identity** specified.

## Exceptions

[ArgumentException](#)

The **identity** does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## ReadInt64(PlcIdentity, Int32)

Reads a [Int64](#) array (in PLC an ARRAY OF DINT) which is associated with the [identity](#) specified.

**C#**

```
public long[] ReadInt64(PlcIdentity identity, int count)
```

### Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

[count](#) [Int32](#)

The number of items to read.

### Returns

[Int64\[\]](#)

The array stored at the data area identified by the [identity](#) specified.

### Exceptions

[ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadInt64(PlcInt64ArrayType)

Reads a [Int64](#) array (in PLC an ARRAY OF DINT) using the [type](#) specified.

**C#**

```
public long[] ReadInt64(PlcInt64ArrayType type)
```

## Parameters

[type](#) [PlcInt64ArrayType](#)

The [PlcInt64ArrayType](#) which addresses the data area to read.

## Returns

[Int64\[\]](#)

The array stored at the data area described by the [type](#) specified.

## Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadInt64(PlcInt64Type)

Reads a [Int64](#) (in PLC a DINT) using the [type](#) specified.

**C#**

```
public long ReadInt64(PlcInt64Type type)
```

## Parameters

[type](#) [PlcInt64Type](#)

The [PlcInt64Type](#) which addresses the data area to read.

## Returns

[Int64](#)

The value stored at the data area described by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# ReadInt64(String)

Reads a `Int64` (in PLC a DINT) which is associated with the `identity` specified.

## C#

```
public long ReadInt64(string identity)
```

## Parameters

### `identity` String

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to read. Or a `PlcName` representing the symbolic name of the `PlcType` (to read) associated with the `Device` of this connection.

## Returns

### Int64

The value stored at the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` is an empty string or does not refer to a `PlcType` that addresses data of the raw type `DWord`.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# ReadInt64(String, Int32)

Reads a [Int64](#) array (in PLC an ARRAY OF DINT) which is associated with the [identity](#) specified.

## C#

```
public long[] ReadInt64(string identity, int count)
```

## Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

[count](#) [Int32](#)

The number of items to read.

## Returns

[Int64\[\]](#)

The array stored at the data area identified by the [identity](#) specified.

## Exceptions

[ArgumentException](#)

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadInt64Core(PlcType)

Reads the data area described by the **type** specified.

**C#**

```
protected virtual long[] ReadInt64Core(PlcType type)
```

## Parameters

**type** PlcType

The PlcInt64Type or PlcInt64ArrayType to read.

## Returns

Int64[]

An array of the Int64 data read.

# ReadLReal(PlcIdentity)

Reads a Double (in PLC a LREAL) which is associated with the **identity** specified.

**C#**

```
public double ReadLReal(PlcIdentity identity)
```

## Parameters

**identity** PlcIdentity

The PlcIdentity that can be either a PlcAddress which addresses the data area to read. Or a PlcName representing the symbolic name of the PlcType (to read) associated with the Device of this connection.

## Returns

Double

The value stored at the data area identified by the **identity** specified.

## Exceptions

ArgumentException

The **identity** does not refer to a PlcType that addresses data of the raw type LWord.

ArgumentNullException

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

InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## ReadLReal(PlcIdentity, Int32)

Reads a [Double](#) array (in PLC an ARRAY OF LREAL) which is associated with the [identity](#) specified.

**C#**

```
public double[] ReadLReal(PlcIdentity identity, int count)
```

### Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

[count](#) [Int32](#)

The number of items to read.

### Returns

[Double\[\]](#)

The array stored at the data area identified by the [identity](#) specified.

### Exceptions

[ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [LWord](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.



# ReadLReal(PlcLRealArrayType)

Reads a [Double](#) array (in PLC an ARRAY OF LREAL) using the [type](#) specified.

**C#**

```
public double[] ReadLReal(PlcLRealArrayType type)
```

## Parameters

[type](#) [PlcLRealArrayType](#)

The [PlcLRealArrayType](#) which addresses the data area to read.

## Returns

[Double\[\]](#)

The array stored at the data area described by the [type](#) specified.

## Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadLReal(PlcLRealType)

Reads a [Double](#) (in PLC a LREAL) using the [type](#) specified.

**C#**

```
public double ReadLReal(PlcLRealType type)
```

## Parameters

[type](#) [PlcLRealType](#)

The [PlcLRealType](#) which addresses the data area to read.

## Returns

[Double](#)

The value stored at the data area described by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# ReadLReal(String)

Reads a `Double` (in PLC a LREAL) which is associated with the `identity` specified.

## C#

```
public double ReadLReal(string identity)
```

## Parameters

### `identity` String

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to read. Or a `PlcName` representing the symbolic name of the `PlcType` (to read) associated with the `Device` of this connection.

## Returns

### Double

The value stored at the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` is an empty string or does not refer to a `PlcType` that addresses data of the raw type `LWord`.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# ReadLReal(String, Int32)

Reads a [Double](#) array (in PLC an ARRAY OF LREAL) which is associated with the [identity](#) specified.

## C#

```
public double[] ReadLReal(string identity, int count)
```

## Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

[count](#) [Int32](#)

The number of items to read.

## Returns

[Double\[\]](#)

The array stored at the data area identified by the [identity](#) specified.

## Exceptions

[ArgumentException](#)

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [LWord](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadLRealCore(PlcType)

Reads the data area described by the **type** specified.

**C#**

```
protected virtual double[] ReadLRealCore(PlcType type)
```

## Parameters

**type** PlcType

The PlcLRealType or PlcLRealArrayType to read.

## Returns

Double[]

An array of the Double data read.

# ReadNodeCore(PlcDataNode)

Reads the data area described by the **node** specified.

**C#**

```
protected virtual object ReadNodeCore(PlcDataNode node)
```

## Parameters

**node** PlcDataNode

The PlcDataNode to read.

## Returns

Object

The data read.

# ReadNodesCore(IEnumerable<PlcDataNode>)

When implemented in a derived class, reads the data areas described by the **nodes** specified.

**C#**

```
protected abstract IEnumerable<object> ReadNodesCore(IEnumerable<PlcDataNode> nodes)
```

## Parameters

**nodes** IEnumerable<PlcDataNode>

The [PlcDataNode](#)'s to read.

## Returns

[IEnumerable](#)<[Object](#)>

The sequence of data read.

# ReadNodesCore(PlcDataNode[])

Reads the data areas described by the [nodes](#) specified.

## C#

```
protected IEnumerable<object> ReadNodesCore(params PlcDataNode[] nodes)
```

## Parameters

[nodes](#) [PlcDataNode](#)[]

The [PlcDataNode](#)'s to read.

## Returns

[IEnumerable](#)<[Object](#)>

The sequence of data read.

# ReadObject(PlcIdentity)

Reads a [PlcObject](#) (in PLC a STRUCT) which is associated with the [identity](#) specified.

## C#

```
public PlcObject ReadObject(PlcIdentity identity)
```

## Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[PlcObject](#)

The value stored at the data area identified by the [identity](#) specified.

## Exceptions

### ArgumentException

The **identity** does not refer to a **PlcType** that addresses data of the raw type **Byte**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadObject(PlcObjectType)

Reads a **PlcObject** (in PLC a STRUCT) using the **type** specified.

### C#

```
public PlcObject ReadObject(PlcObjectType type)
```

### Parameters

**type** **PlcObjectType**

The **PlcObjectType** which addresses the data area to read.

### Returns

**PlcObject**

The value stored at the data area described by the **type** specified.

### Exceptions

#### ArgumentNullException

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

#### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

# ReadObject(String)

Reads a [PlcObject](#) (in PLC a STRUCT) which is associated with the [identity](#) specified.

**C#**

```
public PlcObject ReadObject(string identity)
```

## Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[PlcObject](#)

The value stored at the data area identified by the [identity](#) specified.

## Exceptions

[ArgumentException](#)

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [Byte](#).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadObject<T>()

Reads the [PlcObject](#) specified by [T](#).

**C#**

```
public T ReadObject<T>()  
    where T : PlcObject, new()
```

## Returns

[T](#)

A new instance of the [PlcObject](#) type specified by [T](#) initialized with the values stored in the PLC.

## Exceptions

### InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadObject<T>(T)

Reads the [PlcObject](#) specified by [T](#) using the [instance](#) passed.

### C#

```
public T ReadObject<T>(T instance)
    where T : PlcObject
```

### Parameters

[instance](#) T

The instance of [T](#) to read.

### Returns

[T](#)

The [instance](#) its [Members](#) where read from the PLC.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadReal(PlcIdentity)

Reads a [Single](#) (in PLC a REAL) which is associated with the [identity](#) specified.

### C#

```
public float ReadReal(PlcIdentity identity)
```



## Parameters

**identity** [PlcIdentity](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[Single](#)

The value stored at the data area identified by the **identity** specified.

## Exceptions

[ArgumentException](#)

The **identity** does not refer to a [PlcType](#) that addresses data of the raw type [LWord](#).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadReal(PlcIdentity, Int32)

Reads a [Single](#) array (in PLC an ARRAY OF REAL) which is associated with the **identity** specified.

## C#

```
public float[] ReadReal(PlcIdentity identity, int count)
```

## Parameters

**identity** [PlcIdentity](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

**count** [Int32](#)

The number of items to read.

## Returns

[Single\[\]](#)

The array stored at the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` does not refer to a `PlcType` that addresses data of the raw type `LWord`.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The `count` is outside of the bounds defined by `MinLength` and `MaxLength`.

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# ReadReal(PlcRealArrayType)

Reads a `Single` array (in PLC an ARRAY OF REAL) using the `type` specified.

## C#

```
public float[] ReadReal(PlcRealArrayType type)
```

## Parameters

### `type PlcRealArrayType`

The `PlcRealArrayType` which addresses the data area to read.

## Returns

### `Single[]`

The array stored at the data area described by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the

connection is in **Faulted** state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## ReadReal(PlcRealType)

Reads a **Single** (in PLC a REAL) using the **type** specified.

**C#**

```
public float ReadReal(PlcRealType type)
```

### Parameters

**type** **PlcRealType**

The **PlcRealType** which addresses the data area to read.

### Returns

**Single**

The value stored at the data area described by the **type** specified.

### Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## ReadReal(String)

Reads a **Single** (in PLC a REAL) which is associated with the **identity** specified.

**C#**

```
public float ReadReal(string identity)
```

### Parameters

**identity** **String**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to read. Or a **PlcName** representing the symbolic name of the **PlcType** (to read) associated with the **Device** of this connection.

## Returns

**Single**

The value stored at the data area identified by the **identity** specified.

## Exceptions

**ArgumentException**

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **LWord**.

**ArgumentNullException**

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

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# ReadReal(String, Int32)

Reads a **Single** array (in PLC an ARRAY OF REAL) which is associated with the **identity** specified.

**C#**

```
public float[] ReadReal(string identity, int count)
```

## Parameters

**identity** **String**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to read. Or a **PlcName** representing the symbolic name of the **PlcType** (to read) associated with the **Device** of this connection.

**count** **Int32**

The number of items to read.

## Returns

**Single[]**

The array stored at the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **LWord**.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The **count** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadRealCore(PlcType)

Reads the data area described by the **type** specified.

### C#

```
protected virtual float[] ReadRealCore(PlcType type)
```

### Parameters

**type** **PlcType**

The **PlcRealType** or **PlcRealArrayType** to read.

### Returns

**Single[]**

An array of the **Single** data read.

## ReadS5Time(PlcIdentity)

Reads a **TimeSpan** (in PLC a S5TIME) which is associated with the **identity** specified.

### C#

```
public TimeSpan ReadS5Time(PlcIdentity identity)
```

### Parameters

**identity** **PlcIdentity**

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[TimeSpan](#)

The value stored at the data area identified by the [identity](#) specified.

## Exceptions

[ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [Word](#).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadS5Time(PlcS5TimeType)

Reads a [TimeSpan](#) (in PLC a S5TIME) using the [type](#) specified.

## C#

```
public TimeSpan ReadS5Time(PlcS5TimeType type)
```

## Parameters

[type](#) [PlcS5TimeType](#)

The [PlcS5TimeType](#) which addresses the data area to read.

## Returns

[TimeSpan](#)

The value stored at the data area described by the [type](#) specified.

## Exceptions

[ArgumentNullException](#)

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

#### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## ReadS5Time(String)

Reads a `TimeSpan` (in PLC a S5TIME) which is associated with the `identity` specified.

### C#

```
public TimeSpan ReadS5Time(string identity)
```

### Parameters

#### `identity` String

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to read. Or a `PlcName` representing the symbolic name of the `PlcType` (to read) associated with the `Device` of this connection.

### Returns

#### `TimeSpan`

The value stored at the data area identified by the `identity` specified.

### Exceptions

#### ArgumentException

The `identity` is an empty string or does not refer to a `PlcType` that addresses data of the raw type `Word`.

#### ArgumentNullException

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

#### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

# ReadS5TimeCore(PlcType)

Reads the data area described by the **type** specified.

**C#**

```
protected virtual TimeSpan ReadS5TimeCore(PlcType type)
```

## Parameters

**type** PlcType

The **PlcS5TimeType** to read.

## Returns

TimeSpan

The **TimeSpan** data read.

# ReadString(PlcIdentity)

Reads a **String** (in PLC a STRING) which is associated with the **identity** specified.

**C#**

```
public string ReadString(PlcIdentity identity)
```

## Parameters

**identity** PlcIdentity

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to read. Or a **PlcName** representing the symbolic name of the **PlcType** (to read) associated with the **Device** of this connection.

## Returns

String

The value stored at the data area identified by the **identity** specified.

## Exceptions

**ArgumentException**

The **identity** does not refer to a **PlcType** that addresses data of the raw type **Byte**.

**ArgumentNullException**

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

**InvalidOperationException**



The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

#### [ObjectDisposedException](#)

The connection has been disposed of.

## ReadString(PlcIdentity, Int32)

Reads a [String](#) (in PLC a STRING) which is associated with the [identity](#) specified.

### C#

```
public string ReadString(PlcIdentity identity, int length)
```

### Parameters

#### [identity](#) PlcIdentity

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

#### [length](#) Int32

The number of characters to read.

### Returns

#### [String](#)

The [String](#) stored at the data area identified by the [identity](#) specified.

### Exceptions

#### [ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [Byte](#).

#### [ArgumentNullException](#)

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

#### [ArgumentOutOfRangeException](#)

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

#### [InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

#### [ObjectDisposedException](#)

The connection has been disposed of.

# ReadString(PlcStringType)

Reads a [String](#) (in PLC a STRING) using the [type](#) specified.

**C#**

```
public string ReadString(PlcStringType type)
```

## Parameters

[type](#) [PlcStringType](#)

The [PlcStringType](#) which addresses the data area to read.

## Returns

[String](#)

The value stored at the data area described by the [type](#) specified.

## Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadString(String)

Reads a [String](#) (in PLC a STRING) which is associated with the [identity](#) specified.

**C#**

```
public string ReadString(string identity)
```

## Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[String](#)

The value stored at the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` is an empty string or does not refer to a `PlcType` that addresses data of the raw type `Byte`.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# ReadString(String, Int32)

Reads a `String` (in PLC a `STRING`) which is associated with the `identity` specified.

## C#

```
public string ReadString(string identity, int length)
```

## Parameters

### identity String

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to read. Or a `PlcName` representing the symbolic name of the `PlcType` (to read) associated with the `Device` of this connection.

### length Int32

The number of characters to read.

## Returns

### String

The `String` stored at the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` is an empty string or does not refer to a `PlcType` that addresses data of the raw type `Byte`.

### ArgumentNullException

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

#### ArgumentOutOfRangeException

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

#### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## ReadStringCore(PlcType)

Reads the data area described by the `type` specified.

### C#

```
protected virtual string ReadStringCore(PlcType type)
```

### Parameters

`type` `PlcType`

The `PlcStringType` to read.

### Returns

`String`

The `String` data read.

## ReadTime(PlcIdentity)

Reads a `TimeSpan` (in PLC a `TIME_OF_DAY`) which is associated with the `identity` specified.

### C#

```
public TimeSpan ReadTime(PlcIdentity identity)
```

### Parameters

`identity` `PlcIdentity`

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to read. Or a `PlcName` representing the symbolic name of the `PlcType` (to read) associated with the `Device` of this connection.

### Returns

`TimeSpan`

The value stored at the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` does not refer to a `PlcType` that addresses data of the raw type `DWord`.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# ReadTime(PlcTimeType)

Reads a `TimeSpan` (in PLC a `TIME_OF_DAY`) using the `type` specified.

## C#

```
public TimeSpan ReadTime(PlcTimeType type)
```

## Parameters

### `type PlcTimeType`

The `PlcTimeType` which addresses the data area to read.

## Returns

### `TimeSpan`

The value stored at the data area described by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadTime(String)

Reads a [TimeSpan](#) (in PLC a TIME\_OF\_DAY) which is associated with the [identity](#) specified.

**C#**

```
public TimeSpan ReadTime(string identity)
```

### Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

### Returns

[TimeSpan](#)

The value stored at the data area identified by the [identity](#) specified.

### Exceptions

[ArgumentException](#)

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## ReadTimeCore(PlcType)

Reads the data area described by the [type](#) specified.

**C#**

```
protected virtual TimeSpan ReadTimeCore(PlcType type)
```

### Parameters

type [PlcType](#)

The [PlcTimeType](#) to read.

## Returns

[TimeSpan](#)

The [TimeSpan](#) data read.

# ReadTimeOfDay(PlcIdentity)

Reads a [TimeSpan](#) (in PLC a TIME\_OF\_DAY) which is associated with the [identity](#) specified.

## C#

```
public TimeSpan ReadTimeOfDay(PlcIdentity identity)
```

## Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[TimeSpan](#)

The value stored at the data area identified by the [identity](#) specified.

## Exceptions

[ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadTimeOfDay(PlcTimeOfDayType)

Reads a [TimeSpan](#) (in PLC a TIME\_OF\_DAY) using the [type](#) specified.

**C#**

```
public TimeSpan ReadTimeOfDay(PlcTimeOfDayType type)
```

## Parameters

[type](#) [PlcTimeOfDayType](#)

The [PlcTimeOfDayType](#) which addresses the data area to read.

## Returns

[TimeSpan](#)

The value stored at the data area described by the [type](#) specified.

## Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadTimeOfDay(String)

Reads a [TimeSpan](#) (in PLC a TIME\_OF\_DAY) which is associated with the [identity](#) specified.

**C#**

```
public TimeSpan ReadTimeOfDay(string identity)
```

## Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[TimeSpan](#)



The value stored at the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **DWord**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadTimeOfDayCore(PlcType)

Reads the data area described by the **type** specified.

### C#

```
protected virtual TimeSpan ReadTimeOfDayCore(PlcType type)
```

## Parameters

**type** **PlcType**

The **PlcTimeOfDayType** to read.

## Returns

**TimeSpan**

The **TimeSpan** data read.

## ReadUInt16(PlcIdentity)

Reads a **UInt16** (in PLC a WORD) which is associated with the **identity** specified.

### C#

```
[CLSCompliant(false)]
public ushort ReadUInt16(PlcIdentity identity)
```

## Parameters

**identity** **PlcIdentity**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to read. Or a **PlcName** representing the symbolic name of the **PlcType** (to read) associated with the **Device** of this connection.

## Returns

**UInt16**

The value stored at the data area identified by the **identity** specified.

## Exceptions

**ArgumentException**

The **identity** does not refer to a **PlcType** that addresses data of the raw type **Word**.

**ArgumentNullException**

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

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# ReadUInt16(PlcIdentity, Int32)

Reads a **UInt16** array (in PLC an ARRAY OF WORD) which is associated with the **identity** specified.

## C#

```
[CLSCompliant(false)]
public ushort[] ReadUInt16(PlcIdentity identity, int count)
```

## Parameters

**identity** **PlcIdentity**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to read. Or a **PlcName** representing the symbolic name of the **PlcType** (to read) associated with the **Device** of this connection.

**count** **Int32**

The number of items to read.

## Returns

**UInt16[]**

The array stored at the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The `identity` does not refer to a `PlcType` that addresses data of the raw type `Word`.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The `count` is outside of the bounds defined by `MinLength` and `MaxLength`.

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadUInt16(PlcUInt16ArrayType)

Reads a `UInt16` array (in PLC an ARRAY OF WORD) using the `type` specified.

### C#

```
[CLSCompliant(false)]
public ushort[] ReadUInt16(PlcUInt16ArrayType type)
```

## Parameters

`type` `PlcUInt16ArrayType`

The `PlcUInt16ArrayType` which addresses the data area to read.

## Returns

`UInt16[]`

The array stored at the data area described by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadUInt16(PlcUInt16Type)

Reads a [UInt16](#) (in PLC a WORD) using the `type` specified.

### C#

```
[CLSCompliant(false)]  
public ushort ReadUInt16(PlcUInt16Type type)
```

### Parameters

`type` [PlcUInt16Type](#)

The [PlcUInt16Type](#) which addresses the data area to read.

### Returns

[UInt16](#)

The value stored at the data area described by the `type` specified.

### Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## ReadUInt16(String)

Reads a [UInt16](#) (in PLC a WORD) which is associated with the `identity` specified.

### C#

```
[CLSCompliant(false)]  
public ushort ReadUInt16(string identity)
```

### Parameters

`identity` [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#)

representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

### UInt16

The value stored at the data area identified by the [identity](#) specified.

## Exceptions

### ArgumentException

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [Word](#).

### ArgumentNullException

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

### InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# ReadUInt16(String, Int32)

Reads a [UInt16](#) array (in PLC an ARRAY OF WORD) which is associated with the [identity](#) specified.

## C#

```
[CLSCompliant(false)]  
public ushort[] ReadUInt16(string identity, int count)
```

## Parameters

### identity String

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

### count Int32

The number of items to read.

## Returns

### UInt16[]

The array stored at the data area identified by the [identity](#) specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Word**.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The **count** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadUInt16Core(PlcType)

Reads the data area described by the **type** specified.

### C#

```
[CLSCompliant(false)]
protected virtual ushort[] ReadUInt16Core(PlcType type)
```

### Parameters

**type** **PlcType**

The **PlcUInt16Type** or **PlcUInt16ArrayType** to read.

### Returns

**UInt16[]**

An array of the **UInt16** data read.

## ReadUInt32(PlcIdentity)

Reads a **UInt32** (in PLC a **DWORD**) which is associated with the **identity** specified.

### C#

```
[CLSCompliant(false)]
public uint ReadUInt32(PlcIdentity identity)
```

### Parameters

## identity PlcIdentity

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to read. Or a **PlcName** representing the symbolic name of the **PlcType** (to read) associated with the **Device** of this connection.

## Returns

UInt32

The value stored at the data area identified by the **identity** specified.

## Exceptions

ArgumentException

The **identity** does not refer to a **PlcType** that addresses data of the raw type **DWord**.

ArgumentNullException

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

InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

ObjectDisposedException

The connection has been disposed of.

# ReadUInt32(PlcIdentity, Int32)

Reads a **UInt32** array (in PLC an ARRAY OF DWORD) which is associated with the **identity** specified.

## C#

```
[CLSCompliant(false)]
public uint[] ReadUInt32(PlcIdentity identity, int count)
```

## Parameters

identity PlcIdentity

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to read. Or a **PlcName** representing the symbolic name of the **PlcType** (to read) associated with the **Device** of this connection.

count Int32

The number of items to read.

## Returns

UInt32[]

The array stored at the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The `identity` does not refer to a `PlcType` that addresses data of the raw type `DWord`.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The `count` is outside of the bounds defined by `MinLength` and `MaxLength`.

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadUInt32(PlcUInt32ArrayType)

Reads a `UInt32` array (in PLC an ARRAY OF DWORD) using the `type` specified.

### C#

```
[CLSCompliant(false)]
public uint[] ReadUInt32(PlcUInt32ArrayType type)
```

### Parameters

`type` `PlcUInt32ArrayType`

The `PlcUInt32ArrayType` which addresses the data area to read.

### Returns

`UInt32[]`

The array stored at the data area described by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException



The connection has been disposed of.

## ReadUInt32(PlcUInt32Type)

Reads a [UInt32](#) (in PLC a DWORD) using the [type](#) specified.

### C#

```
[CLSCompliant(false)]  
public uint ReadUInt32(PlcUInt32Type type)
```

### Parameters

[type](#) [PlcUInt32Type](#)

The [PlcUInt32Type](#) which addresses the data area to read.

### Returns

[UInt32](#)

The value stored at the data area described by the [type](#) specified.

### Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## ReadUInt32(String)

Reads a [UInt32](#) (in PLC a DWORD) which is associated with the [identity](#) specified.

### C#

```
[CLSCompliant(false)]  
public uint ReadUInt32(string identity)
```

### Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#)

representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[UInt32](#)

The value stored at the data area identified by the [identity](#) specified.

## Exceptions

[ArgumentException](#)

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadUInt32(String, Int32)

Reads a [UInt32](#) array (in PLC an ARRAY OF DWORD) which is associated with the [identity](#) specified.

## C#

```
[CLSCompliant(false)]
public uint[] ReadUInt32(string identity, int count)
```

## Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

[count](#) [Int32](#)

The number of items to read.

## Returns

[UInt32\[\]](#)

The array stored at the data area identified by the [identity](#) specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **DWord**.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The **count** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadUInt32Core(PlcType)

Reads the data area described by the **type** specified.

### C#

```
[CLSCompliant(false)]
protected virtual uint[] ReadUInt32Core(PlcType type)
```

### Parameters

**type** **PlcType**

The **PlcUInt32Type** or **PlcUInt32ArrayType** to read.

### Returns

**UInt32[]**

An array of the **UInt32** data read.

## ReadUInt64(PlcIdentity)

Reads a **UInt64** (in PLC a **LWORD**) which is associated with the **identity** specified.

### C#

```
[CLSCompliant(false)]
public ulong ReadUInt64(PlcIdentity identity)
```

### Parameters

## identity PlcIdentity

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to read. Or a **PlcName** representing the symbolic name of the **PlcType** (to read) associated with the **Device** of this connection.

## Returns

### UInt64

The value stored at the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** does not refer to a **PlcType** that addresses data of the raw type **LWord**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# ReadUInt64(PlcIdentity, Int32)

Reads a **UInt64** array (in PLC an ARRAY OF LWORD) which is associated with the **identity** specified.

## C#

```
[CLSCompliant(false)]  
public ulong[] ReadUInt64(PlcIdentity identity, int count)
```

## Parameters

### identity PlcIdentity

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to read. Or a **PlcName** representing the symbolic name of the **PlcType** (to read) associated with the **Device** of this connection.

### count Int32

The number of items to read.

## Returns

### UInt64[]

The array stored at the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The `identity` does not refer to a `PlcType` that addresses data of the raw type `LWord`.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The `count` is outside of the bounds defined by `MinLength` and `MaxLength`.

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadUInt64(PlcUInt64ArrayType)

Reads a `UInt64` array (in PLC an ARRAY OF LWORD) using the `type` specified.

### C#

```
[CLSCompliant(false)]
public ulong[] ReadUInt64(PlcUInt64ArrayType type)
```

### Parameters

`type` `PlcUInt64ArrayType`

The `PlcUInt64ArrayType` which addresses the data area to read.

### Returns

`UInt64[]`

The array stored at the data area described by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadUInt64(PlcUInt64Type)

Reads a [UInt64](#) (in PLC a LWORD) using the [type](#) specified.

### C#

```
[CLSCompliant(false)]  
public ulong ReadUInt64(PlcUInt64Type type)
```

### Parameters

[type](#) [PlcUInt64Type](#)

The [PlcUInt64Type](#) which addresses the data area to read.

### Returns

[UInt64](#)

The value stored at the data area described by the [type](#) specified.

### Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## ReadUInt64(String)

Reads a [UInt64](#) (in PLC a LWORD) which is associated with the [identity](#) specified.

### C#

```
[CLSCompliant(false)]  
public ulong ReadUInt64(string identity)
```

### Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#)

representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[UInt64](#)

The value stored at the data area identified by the [identity](#) specified.

## Exceptions

[ArgumentException](#)

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [LWord](#).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadUInt64(String, Int32)

Reads a [UInt64](#) array (in PLC an ARRAY OF LWORD) which is associated with the [identity](#) specified.

## C#

```
[CLSCompliant(false)]
public ulong[] ReadUInt64(string identity, int count)
```

## Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

[count](#) [Int32](#)

The number of items to read.

## Returns

[UInt64\[\]](#)

The array stored at the data area identified by the [identity](#) specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **LWord**.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The **count** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadUInt64Core(PlcType)

Reads the data area described by the **type** specified.

### C#

```
[CLSCompliant(false)]
protected virtual ulong[] ReadUInt64Core(PlcType type)
```

### Parameters

**type** **PlcType**

The **PlcUInt64Type** or **PlcUInt64ArrayType** to read.

### Returns

**UInt64[]**

An array of the **UInt64** data read.

## ReadValue<T>(IPlcValue<T>)

Reads the specified **value**.

### C#

```
public T ReadValue<T>(IPlcValue<T> value)
```

### Parameters

**value** **IPlcValue<T>**



The `IPlcValue`1` to read.

## Returns

`T`

The framework value of the value specified.

## Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadValue<T>(PlcIdentity)

Reads the `T` which is associated with the `identity` specified.

**C#**

```
public T ReadValue<T>(PlcIdentity identity)
```

## Parameters

`identity` [PlcIdentity](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

`T`

The value stored at the data area identified by the `identity` specified.

## Exceptions

[ArgumentException](#)

The metadata specified could not be resolved to a valid [PlcValueType](#) or the `identity` does not refer to a [PlcType](#) that can be used to access the framework type specified by `T`.

[ArgumentNullException](#)

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

#### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## ReadValue<T>(PlcIdentity, Int32)

Reads an array of the `T` which is associated with the `identity` specified.

### C#

```
public T[] ReadValue<T>(PlcIdentity identity, int count)
    where T : new()
```

### Parameters

`identity` `PlcIdentity`

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to read. Or a `PlcName` representing the symbolic name of the `PlcType` (to read) associated with the `Device` of this connection.

`count` `Int32`

The number of items to read.

### Returns

`T`

The array stored at the data area identified by the `identity` specified.

### Exceptions

#### ArgumentException

The metadata specified could not be resolved to a valid `PlcArrayType` or the `identity` does not refer to a `PlcType` that can be used to access the framework type specified by `T`.

#### ArgumentNullException

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

#### ArgumentOutOfRangeException

The `count` is outside of the bounds defined by `MinLength` and `MaxLength`.

#### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# ReadValue<T>(String)

Reads the **T** which is associated with the **identity** specified.

## C#

```
public T ReadValue<T>(string identity)
```

## Parameters

**identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to read. Or a **PlcName** representing the symbolic name of the **PlcType** (to read) associated with the **Device** of this connection.

## Returns

**T**

The value stored at the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that can be used to access the framework type specified by **T**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# ReadValue<T>(String, Int32)

Reads an array of the **T** which is associated with the **identity** specified.

## C#

```
public T[] ReadValue<T>(string identity, int count)
    where T : new()
```

## Parameters

**identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to read. Or a **PlcName** representing the symbolic name of the **PlcType** (to read) associated with the **Device** of this connection.

**count** Int32

The number of items to read.

## Returns

T

The array stored at the data area identified by the **identity** specified.

## Exceptions

**ArgumentException**

The **identity** is an empty string or does not refer to a **PlcType** that can be used to access the framework type specified by T.

**ArgumentNullException**

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

**ArgumentOutOfRangeException**

The **count** is outside of the bounds defined by **MinLength** and **MaxLength**.

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# ReadValueCore<T>(PlcArrayType)

Reads the data area described by the **type** specified.

**C#**

```
protected virtual T[] ReadValueCore<T>(PlcArrayType type)
```

## Parameters

**type** PlcArrayType

The **PlcArrayType** to read.

## Returns

T

The T array data read.

# ReadValueCore<T>(PlcValueType)

Reads the data area described by the type specified.

C#

```
protected virtual T ReadValueCore<T>(PlcValueType type)
```

## Parameters

type PlcValueType

The PlcValueType to read.

## Returns

T

The T value data read.

# ReadValues(IEnumerable<IPlcValue>)

Reads the values specified.

C#

```
public object[] ReadValues(IEnumerable<IPlcValue> values)
```

## Parameters

values IEnumerable<IPlcValue>

The IPlcValue objects to read.

## Returns

Object[]

An array of framework values of the values specified.

## Exceptions

ArgumentNullException

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

### InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadValues(IEnumerable<PlcIdentity>)

Reads the values associated with the [identities](#) specified.

### C#

```
public object[] ReadValues(IEnumerable<PlcIdentity> identities)
```

### Parameters

[identities](#) [IEnumerable<PlcIdentity>](#)

The [PlcIdentity](#)'s that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

### Returns

[Object\[\]](#)

An array of the values read.

### Exceptions

#### [ArgumentException](#)

It is not possible to read data using a relative type (see [IsAbsolute](#)).

#### [ArgumentNullException](#)

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

#### [InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

#### [ObjectDisposedException](#)

The connection has been disposed of.

## ReadValues(IEnumerable<PlcType>)

Reads the values associated with the [types](#) specified.

### C#

```
public object[] ReadValues(IEnumerable<PlcType> types)
```

## Parameters

**types** [IEnumerable<PlcType>](#)

The [PlcType](#)'s its values are to be read.

## Returns

[Object\[\]](#)

An array of the values read.

## Exceptions

[ArgumentException](#)

It is not possible to read data using a relative type (see [IsAbsolute](#)).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadValues(IEnumerable<String>)

Reads the values associated with the **identities** specified.

## C#

```
public object[] ReadValues(IEnumerable<string> identities)
```

## Parameters

**identities** [IEnumerable<String>](#)

The [PlcIdentity](#)'s that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[Object\[\]](#)

An array of the values read.

## Exceptions

### ArgumentException

One of the **identities** specified is an empty string or it is not possible to read data using a relative type (see **IsAbsolute**).

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# ReadValues(IPlcValue[])

Reads the values specified.

## C#

```
public object[] ReadValues(params IPlcValue[] values)
```

## Parameters

**values** IPlcValue[]

The **IPlcValue** objects to read.

## Returns

**Object[]**

An array of framework values of the values specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.



# ReadValues(PlcIdentity[])

Reads the values associated with the **identities** specified.

**C#**

```
public object[] ReadValues(params PlcIdentity[] identities)
```

## Parameters

**identities** [PlcIdentity\[\]](#)

The [PlcIdentity](#)'s that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[Object\[\]](#)

An array of the values read.

## Exceptions

[ArgumentException](#)

It is not possible to read data using a relative type (see [IsAbsolute](#)).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadValues(PlcType[])

Reads the values associated with the **types** specified.

**C#**

```
public object[] ReadValues(params PlcType[] types)
```

## Parameters

**types** [PlcType\[\]](#)

The [PlcType](#)'s its values are to be read.

## Returns

[Object\[\]](#)

An array of the values read.

## Exceptions

[ArgumentException](#)

It is not possible to read data using a relative type (see [IsAbsolute](#)).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadValues(String[])

Reads the values associated with the [identities](#) specified.

**C#**

```
public object[] ReadValues(params string[] identities)
```

## Parameters

[identities](#) [String\[\]](#)

The [PlcIdentity](#)'s that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

## Returns

[Object\[\]](#)

An array of the values read.

## Exceptions

[ArgumentException](#)

One of the [identities](#) specified is an empty string or it is not possible to read data using a relative type (see [IsAbsolute](#)).

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadValues<T>(IEnumerable<IPlcValue<T>>)

Reads the values specified.

### C#

```
public T[] ReadValues<T>(IEnumerable<IPlcValue<T>> values)
```

### Parameters

**values** **IEnumerable<IPlcValue>**

The **IPlcValue'1** objects to read.

### Returns

**T**

An array of framework values of the values specified.

### Exceptions

#### ArgumentNullException

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

#### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## ReadValues<T>(IPlcValue<T>[])

Reads the values specified.

### C#

```
public T[] ReadValues<T>(params IPlcValue<T>[] values)
```

## Parameters

**values** [IPlcValue<T>\[\]](#)

The [IPlcValue](#) objects to read.

## Returns

[T](#)

An array of framework values of the values specified.

## Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# ReadValues<T>(PlcIdentity, Int32)

Reads an array of the [T](#) which is associated with the **identity** specified.

## C#

```
public T[] ReadValues<T>(PlcIdentity identity, int count)
```

## Parameters

**identity** [PlcIdentity](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to read) associated with the [Device](#) of this connection.

**count** [Int32](#)

The number of items to read.

## Returns

[T](#)

The array stored at the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The `identity` does not refer to a `PlcType` that can be used to access the framework type specified by `T`.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The `count` is outside of the bounds defined by `MinLength` and `MaxLength`.

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadValues<T>(String, Int32)

Reads an array of the `T` which is associated with the `identity` specified.

### C#

```
public T[] ReadValues<T>(string identity, int count)
```

### Parameters

#### `identity` String

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to read. Or a `PlcName` representing the symbolic name of the `PlcType` (to read) associated with the `Device` of this connection.

#### `count` Int32

The number of items to read.

### Returns

#### `T`

The array stored at the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` is an empty string or does not refer to a `PlcType` that can be used to access the framework type specified by `T`.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The `count` is outside of the bounds defined by `MinLength` and `MaxLength`.

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## ReadValuesCore(IEnumerable<IPlcValue>)

Reads the data areas described by the `values` specified.

### C#

```
protected virtual object[] ReadValuesCore(IEnumerable<IPlcValue> values)
```

### Parameters

`values` `IEnumerable<IPlcValue>`

The `IPlcValue`'s to read.

### Returns

`Object[]`

An array of objects read.

## ReadValuesCore(IPlcValue[])

Reads the data areas described by the `values` specified.

### C#

```
protected virtual object[] ReadValuesCore(params IPlcValue[] values)
```

### Parameters

`values` `IPlcValue[]`

The `IPlcValue`'s to read.

### Returns

`Object[]`

An array of objects read.

# ReadValuesCore<T>(PlcArrayType)

Reads the data area described by the `type` specified.

**C#**

```
protected virtual T[] ReadValuesCore<T>(PlcArrayType type)
```

## Parameters

`type` [PlcArrayType](#)

The [PlcArrayType](#) to read.

## Returns

`T`

The `T` array data read.

# Write(IEnumerable<KeyValuePair<PlcIdentity, Object>>)

Writes the values associated with the [PlcIdentity](#) objects in the `identityValues` specified.

**C#**

```
public void Write(IEnumerable<KeyValuePair<PlcIdentity, object>> identityValues)
```

## Parameters

`identityValues` [IEnumerable<KeyValuePair>](#)

The [PlcIdentity](#)'s that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection. While the [PlcIdentity](#) represents the key in the pair the value of the identity to write is stored as the value in the pair.

## Exceptions

[ArgumentException](#)

It is not possible to write data using a relative type (see [IsAbsolute](#)).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## Write(IEnumerable<KeyValuePair<PlcType, Object>>)

Writes the values associated with the [PlcType](#) objects in the [typeValues](#) specified.

**C#**

```
public void Write(IEnumerable<KeyValuePair<PlcType, object>> typeValues)
```

### Parameters

[typeValues](#) [IEnumerable<KeyValuePair>](#)

The [PlcType](#)'s its values are to be written. While the [PlcType](#) represents the key in the pair the value of the type to write is stored as the value in the pair.

### Exceptions

[ArgumentException](#)

It is not possible to write data using a relative type (see [IsAbsolute](#)).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## Write(IEnumerable<KeyValuePair<String, Object>>)

Writes the values associated with the identities in the [identityValues](#) specified.

**C#**

```
public void Write(IEnumerable<KeyValuePair<string, object>> identityValues)
```

### Parameters

[identityValues](#) [IEnumerable<KeyValuePair>](#)

The [PlcIdentity](#)'s that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection. While the [PlcIdentity](#) represents the key in the pair the value of the identity to write is stored as the value in the pair.



## Exceptions

### ArgumentException

One of the **identityValues** specified is an empty string or it is not possible to write data using a relative type (see **IsAbsolute**).

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## Write(KeyValuePair<PlcIdentity, Object>[])

Writes the values associated with the **PlcIdentity** objects in the **identityValues** specified.

### C#

```
public void Write(params KeyValuePair<PlcIdentity, object>[] identityValues)
```

## Parameters

**identityValues** **KeyValuePair<PlcIdentity, Object>[]**

The **PlcIdentity**'s that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection. While the **PlcIdentity** represents the key in the pair the value of the identity to write is stored as the value in the pair.

## Exceptions

### ArgumentException

It is not possible to write data using a relative type (see **IsAbsolute**).

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# Write(KeyValuePair<PlcType, Object>[])

Writes the values associated with the [PlcType](#) objects in the [typeValues](#) specified.

**C#**

```
public void Write(params KeyValuePair<PlcType, object>[] typeValues)
```

## Parameters

[typeValues](#) [KeyValuePair<PlcType, Object>\[\]](#)

The [PlcType](#)'s its values are to be written. While the [PlcType](#) represents the key in the pair the value of the type to write is stored as the value in the pair.

## Exceptions

[ArgumentException](#)

It is not possible to write data using a relative type (see [IsAbsolute](#)).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# Write(KeyValuePair<String, Object>[])

Writes the values associated with the identities in the [identityValues](#) specified.

**C#**

```
public void Write(params KeyValuePair<string, object>[] identityValues)
```

## Parameters

[identityValues](#) [KeyValuePair<String, Object>\[\]](#)

The [PlcIdentity](#)'s that can be either a [PlcAddress](#) which addresses the data area to read. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection. While the [PlcIdentity](#) represents the key in the pair the value of the identity to write is stored as the value in the pair.

## Exceptions

### ArgumentException

One of the **identityValues** specified is an empty string or it is not possible to write data using a relative type (see **IsAbsolute**).

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteBoolean(PlcBooleanArrayType, Boolean[])

Writes the **Boolean** array (in PLC an ARRAY OF BOOL) **values** to the data area associated with the **type** specified.

### C#

```
public void WriteBoolean(PlcBooleanArrayType type, params bool[] values)
```

### Parameters

**type** **PlcBooleanArrayType**

The **PlcBooleanArrayType** which addresses the data area to write.

**values** **Boolean[]**

The values to write to the data area identified by the **type** specified.

### Exceptions

#### ArgumentNullException

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

#### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

# WriteBoolean(PlcBooleanArrayType, IEnumerable<Boolean>)

Writes the **Boolean** array (in PLC an ARRAY OF BOOL) **values** to the data area associated with the **type** specified.

## C#

```
public void WriteBoolean(PlcBooleanArrayType type, IEnumerable<bool> values)
```

## Parameters

**type** **PlcBooleanArrayType**

The **PlcBooleanArrayType** which addresses the data area to write.

**values** **IEnumerable<Boolean>**

The values to write to the data area identified by the **type** specified.

## Exceptions

**ArgumentNullException**

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

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# WriteBoolean(PlcBooleanType, Boolean)

Writes the **Boolean** (in PLC a BOOL) **value** using the **type** specified.

## C#

```
public void WriteBoolean(PlcBooleanType type, bool value)
```

## Parameters

**type** **PlcBooleanType**

The **PlcBooleanType** which addresses the data area to write.

**value** **Boolean**

The value to write to the data area identified by the **type** specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteBoolean(PlcIdentity, Boolean)

Writes the `Boolean` (in PLC a `BOOL`) `value` to the data area associated with the `identity` specified.

### C#

```
public void WriteBoolean(PlcIdentity identity, bool value)
```

## Parameters

### `identity` PlcIdentity

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to write. Or a `PlcName` representing the symbolic name of the `PlcType` (to write) associated with the `Device` of this connection.

### `value` Boolean

The value to write to the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` does not refer to a `PlcType` that addresses data of the raw type `Bit`.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteBoolean(PlcIdentity, Boolean[])

Writes the **Boolean** array (in PLC an ARRAY OF BOOL) **values** to the data area associated with the **identity** specified.

**C#**

```
public void WriteBoolean(PlcIdentity identity, params bool[] values)
```

## Parameters

**identity** **PlcIdentity**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** **Boolean[]**

The values to write to the data area identified by the **identity** specified.

## Exceptions

**ArgumentException**

The **identity** does not refer to a **PlcType** that addresses data of the raw type **Bit**.

**ArgumentNullException**

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

**ArgumentOutOfRangeException**

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# WriteBoolean(PlcIdentity, IEnumerable<Boolean>)

Writes the **Boolean** array (in PLC an ARRAY OF BOOL) **values** to the data area associated with the **identity** specified.

**C#**

```
public void WriteBoolean(PlcIdentity identity, IEnumerable<bool> values)
```

## Parameters

**identity** **PlcIdentity**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** **IEnumerable<Boolean>**

The values to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** does not refer to a **PlcType** that addresses data of the raw type **Bit**.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteBoolean(String, Boolean)

Writes the **Boolean** (in PLC a **BOOL**) **value** to the data area associated with the **identity** specified.

## C#

```
public void WriteBoolean(string identity, bool value)
```

## Parameters

**identity** **String**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**value** **Boolean**

The value to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Bit**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteBoolean(String, Boolean[])

Writes the **Boolean** array (in PLC an ARRAY OF BOOL) **values** to the data area associated with the **identity** specified.

### C#

```
public void WriteBoolean(string identity, params bool[] values)
```

### Parameters

#### **identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

#### **values** Boolean[]

The values to write to the data area identified by the **identity** specified.

### Exceptions

#### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Bit**.

#### ArgumentNullException

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

#### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

#### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.



# WriteBoolean(String, IEnumerable<Boolean>)

Writes the **Boolean** array (in PLC an ARRAY OF BOOL) **values** to the data area associated with the **identity** specified.

**C#**

```
public void WriteBoolean(string identity, IEnumerable<bool> values)
```

## Parameters

**identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** IEnumerable<Boolean>

The values to write to the data area identified by the **identity** specified.

## Exceptions

**ArgumentException**

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Bit**.

**ArgumentNullException**

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

**ArgumentOutOfRangeException**

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# WriteBooleanCore(PlcType, Boolean[])

Writes the data area described by the **type** specified.

**C#**

```
protected virtual void WriteBooleanCore(PlcType type, params bool[] values)
```

## Parameters

**type** PlcType

The [PlcBooleanType](#) or [PlcBooleanArrayType](#) to write.

**values** [Boolean\[\]](#)

The array of [Boolean](#) data to write.

## WriteByte(PlcByteArrayType, Byte[])

Writes the [Byte](#) array (in PLC an ARRAY OF BYTE) **values** to the data area associated with the **type** specified.

### C#

```
public void WriteByte(PlcByteArrayType type, params byte[] values)
```

### Parameters

**type** [PlcByteArrayType](#)

The [PlcByteArrayType](#) which addresses the data area to write.

**values** [Byte\[\]](#)

The values to write to the data area identified by the **type** specified.

### Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## WriteByte(PlcByteArrayType, IEnumerable<Byte>)

Writes the [Byte](#) array (in PLC an ARRAY OF BYTE) **values** to the data area associated with the **type** specified.

### C#

```
public void WriteByte(PlcByteArrayType type, IEnumerable<byte> values)
```

### Parameters

**type** [PlcByteArrayType](#)

The `PlcByteArrayType` which addresses the data area to write.

`values IEnumerable<Byte>`

The values to write to the data area identified by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteByte(PlcByteType, Byte)

Writes the `Byte` (in PLC a BYTE) `value` using the `type` specified.

## C#

```
public void WriteByte(PlcByteType type, byte value)
```

## Parameters

`type PlcByteType`

The `PlcByteType` which addresses the data area to write.

`value Byte`

The value to write to the data area identified by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteByte(PlcIdentity, Byte)

Writes the **Byte** (in PLC a BYTE) **value** to the data area associated with the **identity** specified.

**C#**

```
public void WriteByte(PlcIdentity identity, byte value)
```

## Parameters

**identity** **PlcIdentity**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**value** **Byte**

The value to write to the data area identified by the **identity** specified.

## Exceptions

**ArgumentException**

The **identity** does not refer to a **PlcType** that addresses data of the raw type **Byte**.

**ArgumentNullException**

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

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# WriteByte(PlcIdentity, Byte[])

Writes the **Byte** array (in PLC an ARRAY OF BYTE) **values** to the data area associated with the **identity** specified.

**C#**

```
public void WriteByte(PlcIdentity identity, params byte[] values)
```

## Parameters

**identity** **PlcIdentity**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** **Byte[]**

The values to write to the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` does not refer to a `PlcType` that addresses data of the raw type `Byte`.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of `values` is outside of the bounds defined by `MinLength` and `MaxLength`.

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteByte(PlcIdentity, IEnumerable<Byte>)

Writes the `Byte` array (in PLC an ARRAY OF BYTE) `values` to the data area associated with the `identity` specified.

### C#

```
public void WriteByte(PlcIdentity identity, IEnumerable<byte> values)
```

## Parameters

### `identity` `PlcIdentity`

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to write. Or a `PlcName` representing the symbolic name of the `PlcType` (to write) associated with the `Device` of this connection.

### `values` `IEnumerable<Byte>`

The values to write to the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` does not refer to a `PlcType` that addresses data of the raw type `Byte`.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteByte(String, Byte)

Writes the **Byte** (in PLC a BYTE) **value** to the data area associated with the **identity** specified.

### C#

```
public void WriteByte(string identity, byte value)
```

### Parameters

#### **identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

#### **value** Byte

The value to write to the data area identified by the **identity** specified.

### Exceptions

#### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Byte**.

#### ArgumentNullException

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

#### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## WriteByte(String, Byte[])

Writes the **Byte** array (in PLC an ARRAY OF BYTE) **values** to the data area associated with the **identity** specified.

## C#

```
public void WriteByte(string identity, params byte[] values)
```

### Parameters

**identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** Byte[]

The values to write to the data area identified by the **identity** specified.

### Exceptions

**ArgumentException**

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Byte**.

**ArgumentNullException**

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

**ArgumentOutOfRangeException**

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

## WriteByte(String, IEnumerable<Byte>)

Writes the **Byte** array (in PLC an ARRAY OF BYTE) **values** to the data area associated with the **identity** specified.

## C#

```
public void WriteByte(string identity, IEnumerable<byte> values)
```

### Parameters

**identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** IEnumerable<Byte>

The values to write to the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` is an empty string or does not refer to a `PlcType` that addresses data of the raw type `Byte`.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of `values` is outside of the bounds defined by `MinLength` and `MaxLength`.

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteByteCore(PlcType, Byte[])

Writes the data area described by the `type` specified.

### C#

```
protected virtual void WriteByteCore(PlcType type, params byte[] values)
```

## Parameters

`type` `PlcType`

The `PlcByteType` or `PlcByteArrayType` to write.

`values` `Byte[]`

The array of `Byte` data to write.

## WriteChar(PlcCharArrayType, Char[])

Writes the `Char` array (in PLC an ARRAY OF CHAR) `values` to the data area associated with the `type` specified.

### C#

```
public void WriteChar(PlcCharArrayType type, params char[] values)
```

## Parameters



**type** PlcCharArrayType

The **PlcCharArrayType** which addresses the data area to write.

**values** Char[]

The values to write to the data area identified by the **type** specified.

## Exceptions

**ArgumentNullException**

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

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# WriteChar(PlcCharArrayType, IEnumerable<Char>)

Writes the **Char** array (in PLC an ARRAY OF CHAR) **values** to the data area associated with the **type** specified.

**C#**

```
public void WriteChar(PlcCharArrayType type, IEnumerable<char> values)
```

## Parameters

**type** PlcCharArrayType

The **PlcCharArrayType** which addresses the data area to write.

**values** IEnumerable<Char>

The values to write to the data area identified by the **type** specified.

## Exceptions

**ArgumentNullException**

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

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

## WriteChar(PlcCharType, Char)

Writes the **Char** (in PLC a CHAR) **value** using the **type** specified.

**C#**

```
public void WriteChar(PlcCharType type, char value)
```

### Parameters

**type** **PlcCharType**

The **PlcCharType** which addresses the data area to write.

**value** **Char**

The value to write to the data area identified by the **type** specified.

### Exceptions

**ArgumentNullException**

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

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

## WriteChar(PlcIdentity, Char)

Writes the **Char** (in PLC a CHAR) **value** to the data area associated with the **identity** specified.

**C#**

```
public void WriteChar(PlcIdentity identity, char value)
```

### Parameters

**identity** **PlcIdentity**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**value** **Char**

The value to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The `identity` does not refer to a `PlcType` that addresses data of the raw type `Byte`.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteChar(PlcIdentity, Char[])

Writes the `Char` array (in PLC an ARRAY OF CHAR) `values` to the data area associated with the `identity` specified.

### C#

```
public void WriteChar(PlcIdentity identity, params char[] values)
```

## Parameters

### `identity` `PlcIdentity`

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to write. Or a `PlcName` representing the symbolic name of the `PlcType` (to write) associated with the `Device` of this connection.

### `values` `Char[]`

The values to write to the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` does not refer to a `PlcType` that addresses data of the raw type `Byte`.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of `values` is outside of the bounds defined by `MinLength` and `MaxLength`.

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# WriteChar(PlcIdentity, IEnumerable<Char>)

Writes the [Char](#) array (in PLC an ARRAY OF CHAR) [values](#) to the data area associated with the [identity](#) specified.

## C#

```
public void WriteChar(PlcIdentity identity, IEnumerable<char> values)
```

## Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

[values](#) [IEnumerable<Char>](#)

The values to write to the data area identified by the [identity](#) specified.

## Exceptions

### ArgumentException

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [Byte](#).

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of [values](#) is outside of the bounds defined by [MinLength](#) and [MaxLength](#).

### InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteChar(String, Char)

Writes the [Char](#) (in PLC a CHAR) [value](#) to the data area associated with the [identity](#) specified.

## C#

```
public void WriteChar(string identity, char value)
```

## Parameters

### identity String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

### value Char

The value to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Byte**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteChar(String, Char[])

Writes the **Char** array (in PLC an ARRAY OF CHAR) **values** to the data area associated with the **identity** specified.

## C#

```
public void WriteChar(string identity, params char[] values)
```

## Parameters

### identity String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

### values Char[]

The values to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Byte**.

#### ArgumentNullException

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

#### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

#### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## WriteChar(String, IEnumerable<Char>)

Writes the **Char** array (in PLC an ARRAY OF CHAR) **values** to the data area associated with the **identity** specified.

### C#

```
public void WriteChar(string identity, IEnumerable<char> values)
```

### Parameters

#### **identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

#### **values** IEnumerable<Char>

The values to write to the data area identified by the **identity** specified.

### Exceptions

#### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Byte**.

#### ArgumentNullException

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

#### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

#### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

## WriteCharCore(PlcType, Char[])

Writes the data area described by the `type` specified.

### C#

```
protected virtual void WriteCharCore(PlcType type, params char[] values)
```

### Parameters

`type` [PlcType](#)

The [PlcCharType](#) or [PlcCharArrayType](#) to write.

`values` [Char\[\]](#)

The array of [Char](#) data to write.

## WriteCore(IEnumerable<KeyValuePair<PlcType, Object>>)

Writes the values associated with the [PlcType](#) objects in the `typeValues` specified.

### C#

```
protected virtual void WriteCore(IEnumerable<KeyValuePair<PlcType, object>> typeValues)
```

### Parameters

`typeValues` [IEnumerable<KeyValuePair>](#)

The [PlcType](#)'s its values are to be written. While the [PlcType](#) represents the key in the pair the value of the type to write is stored as the value in the pair.

## WriteDate(PlcDateType, DateTime)

Writes the [DateTime](#) (in PLC a DATE) `value` using the `type` specified.

### C#

```
public void WriteDate(PlcDateType type, DateTime value)
```

### Parameters

`type` [PlcDateType](#)

The [PlcDateType](#) which addresses the data area to write.

value DateTime

The value to write to the data area identified by the type specified.

## Exceptions

ArgumentNullException

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

InvalidOperationException

The connection is not in Opened, Connected or Disconnected state and is therefore not ready to use or the connection is in Faulted state and cannot longer be used.

ObjectDisposedException

The connection has been disposed of.

# WriteDate(PlcIdentity, DateTime)

Writes the DateTime (in PLC a DATE) value to the data area associated with the identity specified.

C#

```
public void WriteDate(PlcIdentity identity, DateTime value)
```

## Parameters

identity PlcIdentity

The PlcIdentity that can be either a PlcAddress which addresses the data area to write. Or a PlcName representing the symbolic name of the PlcType (to write) associated with the Device of this connection.

value DateTime

The value to write to the data area identified by the identity specified.

## Exceptions

ArgumentException

The identity does not refer to a PlcType that addresses data of the raw type Word.

ArgumentNullException

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

InvalidOperationException

The connection is not in Opened, Connected or Disconnected state and is therefore not ready to use or the connection is in Faulted state and cannot longer be used.

ObjectDisposedException



The connection has been disposed of.

## WriteDate(String, DateTime)

Writes the [DateTime](#) (in PLC a DATE) [value](#) to the data area associated with the [identity](#) specified.

**C#**

```
public void WriteDate(string identity, DateTime value)
```

### Parameters

[identity](#) String

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

[value](#) DateTime

The value to write to the data area identified by the [identity](#) specified.

### Exceptions

[ArgumentException](#)

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [Word](#).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## WriteDateCore(PlcType, DateTime)

Writes the data area described by the [type](#) specified.

**C#**

```
protected virtual void WriteDateCore(PlcType type, DateTime value)
```

### Parameters

[type](#) PlcType

The [PlcDateType](#) to write.

value DateTime

The DateTime data to write.

## WriteDateTime(PlcDateTimeType, DateTime)

Writes the DateTime (in PLC a DATE\_AND\_TIME) value using the type specified.

C#

```
public void WriteDateTime(PlcDateTimeType type, DateTime value)
```

### Parameters

type PlcDateTimeType

The PlcDateTimeType which addresses the data area to write.

value DateTime

The value to write to the data area identified by the type specified.

### Exceptions

ArgumentNullException

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

InvalidOperationException

The connection is not in Opened, Connected or Disconnected state and is therefore not ready to use or the connection is in Faulted state and cannot longer be used.

ObjectDisposedException

The connection has been disposed of.

## WriteDateTime(PlcIdentity, DateTime)

Writes the DateTime (in PLC a DATE\_AND\_TIME) value to the data area associated with the identity specified.

C#

```
public void WriteDateTime(PlcIdentity identity, DateTime value)
```

### Parameters

identity PlcIdentity

The PlcIdentity that can be either a PlcAddress which addresses the data area to write. Or a PlcName representing the symbolic name of the PlcType (to write) associated with the Device of this connection.

value DateTime

The value to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** does not refer to a **PlcType** that addresses data of the raw type **Byte**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteDateTime(String, DateTime)

Writes the **DateTime** (in PLC a DATE\_AND\_TIME) **value** to the data area associated with the **identity** specified.

## C#

```
public void WriteDateTime(string identity, DateTime value)
```

## Parameters

### identity String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

### value DateTime

The value to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Byte**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

### [ObjectDisposedException](#)

The connection has been disposed of.

## WriteDateTimeCore(PlcType, DateTime)

Writes the data area described by the `type` specified.

### C#

```
protected virtual void WriteDateTimeCore(PlcType type, DateTime value)
```

### Parameters

`type` [PlcType](#)

The [PlcDateTimeType](#) to write.

`value` [DateTime](#)

The [DateTime](#) data to write.

## WriteDouble(PlcDoubleArrayType, Double[])

Writes the [Double](#) array (in PLC an ARRAY OF DOUBLE) `values` to the data area associated with the `type` specified.

### C#

```
public void WriteDouble(PlcDoubleArrayType type, params double[] values)
```

### Parameters

`type` [PlcDoubleArrayType](#)

The [PlcDoubleArrayType](#) which addresses the data area to write.

`values` [Double\[\]](#)

The values to write to the data area identified by the `type` specified.

### Exceptions

#### [ArgumentNullException](#)

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

#### [InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the

connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## WriteDouble(PlcDoubleArrayType, IEnumerable<Double>)

Writes the **Double** array (in PLC an ARRAY OF DOUBLE) **values** to the data area associated with the **type** specified.

#### C#

```
public void WriteDouble(PlcDoubleArrayType type, IEnumerable<double> values)
```

#### Parameters

**type** PlcDoubleArrayType

The **PlcDoubleArrayType** which addresses the data area to write.

**values** IEnumerable<Double>

The values to write to the data area identified by the **type** specified.

#### Exceptions

##### ArgumentNullException

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

##### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

##### ObjectDisposedException

The connection has been disposed of.

## WriteDouble(PlcDoubleType, Double)

Writes the **Double** (in PLC a DOUBLE) **value** using the **type** specified.

#### C#

```
public void WriteDouble(PlcDoubleType type, double value)
```

#### Parameters

**type** PlcDoubleType

The `PlcDoubleType` which addresses the data area to write.

`value Double`

The value to write to the data area identified by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteDouble(PlcIdentity, Double)

Writes the `Double` (in PLC a `DOUBLE`) `value` to the data area associated with the `identity` specified.

## C#

```
public void WriteDouble(PlcIdentity identity, double value)
```

## Parameters

`identity PlcIdentity`

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to write. Or a `PlcName` representing the symbolic name of the `PlcType` (to write) associated with the `Device` of this connection.

`value Double`

The value to write to the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` does not refer to a `PlcType` that addresses data of the raw type `DWord`.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# WriteDouble(PlcIdentity, Double[])

Writes the **Double** array (in PLC an ARRAY OF DOUBLE) **values** to the data area associated with the **identity** specified.

## C#

```
public void WriteDouble(PlcIdentity identity, params double[] values)
```

## Parameters

**identity** **PlcIdentity**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** **Double[]**

The values to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** does not refer to a **PlcType** that addresses data of the raw type **DWord**.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteDouble(PlcIdentity, IEnumerable<Double>)

Writes the **Double** array (in PLC an ARRAY OF DOUBLE) **values** to the data area associated with the **identity** specified.

## C#

```
public void WriteDouble(PlcIdentity identity, IEnumerable<double> values)
```

## Parameters

**identity** [PlcIdentity](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

**values** [IEnumerable<Double>](#)

The values to write to the data area identified by the **identity** specified.

## Exceptions

[ArgumentException](#)

The **identity** does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

The number of **values** is outside of the bounds defined by [MinLength](#) and [MaxLength](#).

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# WriteDouble(String, Double)

Writes the [Double](#) (in PLC a DOUBLE) **value** to the data area associated with the **identity** specified.

**C#**

```
public void WriteDouble(string identity, double value)
```

## Parameters

**identity** [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

**value** [Double](#)

The value to write to the data area identified by the **identity** specified.



## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **DWord**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteDouble(String, Double[])

Writes the **Double** array (in PLC an ARRAY OF DOUBLE) **values** to the data area associated with the **identity** specified.

### C#

```
public void WriteDouble(string identity, params double[] values)
```

## Parameters

### identity String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

### values Double[]

The values to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **DWord**.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# WriteDouble(String, IEnumerable<Double>)

Writes the [Double](#) array (in PLC an ARRAY OF DOUBLE) [values](#) to the data area associated with the [identity](#) specified.

## C#

```
public void WriteDouble(string identity, IEnumerable<double> values)
```

## Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

[values](#) [IEnumerable<Double>](#)

The values to write to the data area identified by the [identity](#) specified.

## Exceptions

### [ArgumentException](#)

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

### [ArgumentNullException](#)

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

### [ArgumentOutOfRangeException](#)

The number of [values](#) is outside of the bounds defined by [MinLength](#) and [MaxLength](#).

### [InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

### [ObjectDisposedException](#)

The connection has been disposed of.

# WriteDoubleCore(PlcType, Double[])

Writes the data area described by the [type](#) specified.

## C#

```
protected virtual void WriteDoubleCore(PlcType type, params double[] values)
```

## Parameters

**type** [PlcType](#)

The [PlcDoubleType](#) or [PlcDoubleArrayType](#) to write.

**values** [Double](#)[]

The array of [Double](#) data to write.

# WriteEntities(IEnumerable<IPlcEntity>)

Writes the **entities** specified.

## C#

```
public IEnumerable<PlcEntityWriteResult> WriteEntities(IEnumerable<IPlcEntity> entities)
```

## Parameters

**entities** [IEnumerable<IPlcEntity>](#)

The [IPlcEntity](#)'s to write.

## Returns

[IEnumerable<PlcEntityWriteResult>](#)

A sequence of [PlcEntityWriteResult](#) instances containing the [IPlcEntity](#) and its resulting data after a write operation has been performed for the entity.

## Exceptions

[ArgumentException](#)

It is not possible to write data using a relative type (see [IsAbsolute](#)).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## Remarks

In case there a [PlcType](#) is passed as an [IPlcEntity](#) the default value of the [PlcType](#) is written.

# WriteEntities(IPlcEntity[])

Writes the **entities** specified.

**C#**

```
public IEnumerable<PlcEntityWriteResult> WriteEntities(params IPlcEntity[] entities)
```

## Parameters

**entities** [IPlcEntity\[\]](#)

The [IPlcEntity](#)'s to write.

## Returns

[IEnumerable<PlcEntityWriteResult>](#)

A sequence of [PlcEntityWriteResult](#) instances containing the [IPlcEntity](#) and its resulting data after a write operation has been performed for the entity.

## Exceptions

[ArgumentException](#)

It is not possible to write data using a relative type (see [IsAbsolute](#)).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## Remarks

In case there a [PlcType](#) is passed as an [IPlcEntity](#) the default value of the [PlcType](#) is written.

# WriteInt16(PlcIdentity, IEnumerable<Int16>)

Writes the [Int16](#) array (in PLC an ARRAY OF INT) **values** to the data area associated with the **identity** specified.

**C#**

```
public void WriteInt16(PlcIdentity identity, IEnumerable<short> values)
```

## Parameters

**identity** PlcIdentity

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** IEnumerable<Int16>

The values to write to the data area identified by the **identity** specified.

## Exceptions

**ArgumentException**

The **identity** does not refer to a **PlcType** that addresses data of the raw type **Word**.

**ArgumentNullException**

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

**ArgumentOutOfRangeException**

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# WriteInt16(PlcIdentity, Int16)

Writes the **Int16** (in PLC a INT) **value** to the data area associated with the **identity** specified.

**C#**

```
public void WriteInt16(PlcIdentity identity, short value)
```

## Parameters

**identity** PlcIdentity

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**value** Int16

The value to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** does not refer to a **PlcType** that addresses data of the raw type **Word**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteInt16(PlcIdentity, Int16[])

Writes the **Int16** array (in PLC an ARRAY OF INT) **values** to the data area associated with the **identity** specified.

### C#

```
public void WriteInt16(PlcIdentity identity, params short[] values)
```

### Parameters

**identity** **PlcIdentity**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** **Int16[]**

The values to write to the data area identified by the **identity** specified.

### Exceptions

#### ArgumentException

The **identity** does not refer to a **PlcType** that addresses data of the raw type **Word**.

#### ArgumentNullException

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

#### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

#### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## WriteInt16(PlcInt16ArrayType, IEnumerable<Int16>)

Writes the **Int16** array (in PLC an ARRAY OF INT) **values** to the data area associated with the **type** specified.

### C#

```
public void WriteInt16(PlcInt16ArrayType type, IEnumerable<short> values)
```

### Parameters

**type** **PlcInt16ArrayType**

The **PlcInt16ArrayType** which addresses the data area to write.

**values** **IEnumerable<Int16>**

The values to write to the data area identified by the **type** specified.

### Exceptions

**ArgumentNullException**

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

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

## WriteInt16(PlcInt16ArrayType, Int16[])

Writes the **Int16** array (in PLC an ARRAY OF INT) **values** to the data area associated with the **type** specified.

### C#

```
public void WriteInt16(PlcInt16ArrayType type, params short[] values)
```

### Parameters

**type** **PlcInt16ArrayType**

The **PlcInt16ArrayType** which addresses the data area to write.

**values** **Int16[]**

The values to write to the data area identified by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteInt16(PlcInt16Type, Int16)

Writes the `Int16` (in PLC a INT) `value` using the `type` specified.

### C#

```
public void WriteInt16(PlcInt16Type type, short value)
```

## Parameters

`type` `PlcInt16Type`

The `PlcInt16Type` which addresses the data area to write.

`value` `Int16`

The value to write to the data area identified by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.



# WriteInt16(String, IEnumerable<Int16>)

Writes the **Int16** array (in PLC an ARRAY OF INT) **values** to the data area associated with the **identity** specified.

**C#**

```
public void WriteInt16(string identity, IEnumerable<short> values)
```

## Parameters

**identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** IEnumerable<Int16>

The values to write to the data area identified by the **identity** specified.

## Exceptions

**ArgumentException**

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Word**.

**ArgumentNullException**

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

**ArgumentOutOfRangeException**

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# WriteInt16(String, Int16)

Writes the **Int16** (in PLC a INT) **value** to the data area associated with the **identity** specified.

**C#**

```
public void WriteInt16(string identity, short value)
```

## Parameters

**identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**value** **Int16**

The value to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Word**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteInt16(String, Int16[])

Writes the **Int16** array (in PLC an ARRAY OF INT) **values** to the data area associated with the **identity** specified.

### C#

```
public void WriteInt16(string identity, params short[] values)
```

## Parameters

**identity** **String**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** **Int16[]**

The values to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Word**.

### ArgumentNullException

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

#### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

#### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## WriteInt16Core(PlcType, Int16[])

Writes the data area described by the **type** specified.

### C#

```
protected virtual void WriteInt16Core(PlcType type, params short[] values)
```

### Parameters

**type** PlcType

The **PlcInt16Type** or **PlcInt16ArrayType** to write.

**values** Int16[]

The array of **Int16** data to write.

## WriteInt32(PlcIdentity, IEnumerable<Int32>)

Writes the **Int32** array (in PLC an ARRAY OF DINT) **values** to the data area associated with the **identity** specified.

### C#

```
public void WriteInt32(PlcIdentity identity, IEnumerable<int> values)
```

### Parameters

**identity** PlcIdentity

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** IEnumerable<Int32>

The values to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** does not refer to a **PlcType** that addresses data of the raw type **DWord**.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteInt32(PlcIdentity, Int32)

Writes the **Int32** (in PLC a DINT) **value** to the data area associated with the **identity** specified.

### C#

```
public void WriteInt32(PlcIdentity identity, int value)
```

## Parameters

### **identity** PlcIdentity

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

### **value** Int32

The value to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** does not refer to a **PlcType** that addresses data of the raw type **DWord**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# WriteInt32(PlcIdentity, Int32[])

Writes the [Int32](#) array (in PLC an ARRAY OF DINT) [values](#) to the data area associated with the [identity](#) specified.

## C#

```
public void WriteInt32(PlcIdentity identity, params int[] values)
```

## Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

[values](#) [Int32\[\]](#)

The values to write to the data area identified by the [identity](#) specified.

## Exceptions

[ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

The number of [values](#) is outside of the bounds defined by [MinLength](#) and [MaxLength](#).

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# WriteInt32(PlcInt32ArrayType, IEnumerable<Int32>)

Writes the [Int32](#) array (in PLC an ARRAY OF DINT) [values](#) to the data area associated with the [type](#) specified.

## C#

```
public void WriteInt32(PlcInt32ArrayType type, IEnumerable<int> values)
```

## Parameters

**type** [PlcInt32ArrayType](#)

The [PlcInt32ArrayType](#) which addresses the data area to write.

**values** [IEnumerable<Int32>](#)

The values to write to the data area identified by the **type** specified.

## Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# WriteInt32(PlcInt32ArrayType, Int32[])

Writes the [Int32](#) array (in PLC an ARRAY OF DINT) **values** to the data area associated with the **type** specified.

## C#

```
public void WriteInt32(PlcInt32ArrayType type, params int[] values)
```

## Parameters

**type** [PlcInt32ArrayType](#)

The [PlcInt32ArrayType](#) which addresses the data area to write.

**values** [Int32\[\]](#)

The values to write to the data area identified by the **type** specified.

## Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## WriteInt32(PlcInt32Type, Int32)

Writes the **Int32** (in PLC a DINT) **value** using the **type** specified.

#### C#

```
public void WriteInt32(PlcInt32Type type, int value)
```

#### Parameters

**type** **PlcInt32Type**

The **PlcInt32Type** which addresses the data area to write.

**value** **Int32**

The value to write to the data area identified by the **type** specified.

#### Exceptions

##### ArgumentNullException

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

##### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

##### ObjectDisposedException

The connection has been disposed of.

## WriteInt32(String, IEnumerable<Int32>)

Writes the **Int32** array (in PLC an ARRAY OF DINT) **values** to the data area associated with the **identity** specified.

#### C#

```
public void WriteInt32(string identity, IEnumerable<int> values)
```

#### Parameters

**identity** **String**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** **IEnumerable<Int32>**

The values to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **DWord**.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteInt32(String, Int32)

Writes the **Int32** (in PLC a DINT) **value** to the data area associated with the **identity** specified.

## C#

```
public void WriteInt32(string identity, int value)
```

## Parameters

**identity** **String**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**value** **Int32**

The value to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **DWord**.



## ArgumentNullException

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

## InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# WriteInt32(String, Int32[])

Writes the **Int32** array (in PLC an ARRAY OF DINT) **values** to the data area associated with the **identity** specified.

## C#

```
public void WriteInt32(string identity, params int[] values)
```

## Parameters

### **identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

### **values** Int32[]

The values to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **DWord**.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteInt32Core(PlcType, Int32[])

Writes the data area described by the `type` specified.

**C#**

```
protected virtual void WriteInt32Core(PlcType type, params int[] values)
```

### Parameters

`type` [PlcType](#)

The [PlcInt32Type](#) or [PlcInt32ArrayType](#) to write.

`values` [Int32\[\]](#)

The array of [Int32](#) data to write.

## WriteInt64(PlcIdentity, IEnumerable<Int64>)

Writes the [Int64](#) array (in PLC an ARRAY OF LINT) `values` to the data area associated with the `identity` specified.

**C#**

```
[CLSCompliant(false)]  
public void WriteInt64(PlcIdentity identity, IEnumerable<long> values)
```

### Parameters

`identity` [PlcIdentity](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

`values` [IEnumerable<Int64>](#)

The values to write to the data area identified by the `identity` specified.

### Exceptions

[ArgumentException](#)

The `identity` does not refer to a [PlcType](#) that addresses data of the raw type [LWord](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

The number of `values` is outside of the bounds defined by [MinLength](#) and [MaxLength](#).

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

#### [ObjectDisposedException](#)

The connection has been disposed of.

## WriteInt64(PlcIdentity, Int64)

Writes the [Int64](#) (in PLC a LINT) [value](#) to the data area associated with the [identity](#) specified.

### C#

```
[CLSCompliant(false)]  
public void WriteInt64(PlcIdentity identity, long value)
```

### Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

[value](#) [Int64](#)

The value to write to the data area identified by the [identity](#) specified.

### Exceptions

#### [ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [LWord](#).

#### [ArgumentNullException](#)

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

#### [InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

#### [ObjectDisposedException](#)

The connection has been disposed of.

## WriteInt64(PlcIdentity, Int64[])

Writes the [Int64](#) array (in PLC an ARRAY OF LINT) [values](#) to the data area associated with the [identity](#) specified.

### C#

```
[CLSCompliant(false)]
public void WriteInt64(PlcIdentity identity, params long[] values)
```

## Parameters

**identity** PlcIdentity

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** Int64[]

The values to write to the data area identified by the **identity** specified.

## Exceptions

**ArgumentException**

The **identity** does not refer to a **PlcType** that addresses data of the raw type **LWord**.

**ArgumentNullException**

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

**ArgumentOutOfRangeException**

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# WriteInt64(PlcInt64ArrayType, IEnumerable<Int64>)

Writes the **Int64** array (in PLC an ARRAY OF LINT) **values** to the data area associated with the **type** specified.

## C#

```
[CLSCompliant(false)]
public void WriteInt64(PlcInt64ArrayType type, IEnumerable<long> values)
```

## Parameters

**type** PlcInt64ArrayType

The **PlcInt64ArrayType** which addresses the data area to write.

**values** IEnumerable<Int64>

The values to write to the data area identified by the **type** specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteInt64(PlcInt64ArrayType, Int64[])

Writes the **Int64** array (in PLC an ARRAY OF LINT) **values** to the data area associated with the **type** specified.

## C#

```
[CLSCompliant(false)]
public void WriteInt64(PlcInt64ArrayType type, params long[] values)
```

## Parameters

**type** **PlcInt64ArrayType**

The **PlcInt64ArrayType** which addresses the data area to write.

**values** **Int64[]**

The values to write to the data area identified by the **type** specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteInt64(PlcInt64Type, Int64)

Writes the **Int64** (in PLC a LINT) **value** using the **type** specified.

## C#

```
[CLSCompliant(false)]  
public void WriteInt64(PlcInt64Type type, long value)
```

## Parameters

**type** **PlcInt64Type**

The **PlcInt64Type** which addresses the data area to write.

**value** **Int64**

The value to write to the data area identified by the **type** specified.

## Exceptions

**ArgumentNullException**

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

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# WriteInt64(String, IEnumerable<Int64>)

Writes the **Int64** array (in PLC an ARRAY OF LINT) **values** to the data area associated with the **identity** specified.

## C#

```
[CLSCompliant(false)]  
public void WriteInt64(string identity, IEnumerable<long> values)
```

## Parameters

**identity** **String**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** **IEnumerable<Int64>**

The values to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **LWord**.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteInt64(String, Int64)

Writes the **Int64** (in PLC a LINT) **value** to the data area associated with the **identity** specified.

### C#

```
[CLSCompliant(false)]
public void WriteInt64(string identity, long value)
```

## Parameters

### **identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

### **value** Int64

The value to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **LWord**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# WriteInt64(String, Int64[])

Writes the [Int64](#) array (in PLC an ARRAY OF LINT) [values](#) to the data area associated with the [identity](#) specified.

## C#

```
[CLSCompliant(false)]  
public void WriteInt64(string identity, params long[] values)
```

## Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

[values](#) [Int64\[\]](#)

The values to write to the data area identified by the [identity](#) specified.

## Exceptions

### [ArgumentException](#)

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [LWord](#).

### [ArgumentNullException](#)

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

### [ArgumentOutOfRangeException](#)

The number of [values](#) is outside of the bounds defined by [MinLength](#) and [MaxLength](#).

### [InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

### [ObjectDisposedException](#)

The connection has been disposed of.

# WriteInt64Core(PlcType, Int64[])

Writes the data area described by the [type](#) specified.

## C#

```
protected virtual void WriteInt64Core(PlcType type, params long[] values)
```



## Parameters

type **PlcType**

The **PlcInt64Type** or **PlcInt64ArrayType** to write.

values **Int64[]**

The array of **Int32** data to write.

# WriteLReal(PlcIdentity, Double)

Writes the **Double** (in PLC a LWORD) **value** to the data area associated with the **identity** specified.

## C#

```
[CLSCompliant(false)]  
public void WriteLReal(PlcIdentity identity, double value)
```

## Parameters

**identity** **PlcIdentity**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**value** **Double**

The value to write to the data area identified by the **identity** specified.

## Exceptions

**ArgumentException**

The **identity** does not refer to a **PlcType** that addresses data of the raw type **LWord**.

**ArgumentNullException**

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

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# WriteLReal(PlcIdentity, Double[])

Writes the **Double** array (in PLC an ARRAY OF LWORD) **values** to the data area associated with the **identity** specified.

## C#

```
[CLSCompliant(false)]
public void WriteLReal(PlcIdentity identity, params double[] values)
```

## Parameters

**identity** [PlcIdentity](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

**values** [Double\[\]](#)

The values to write to the data area identified by the **identity** specified.

## Exceptions

[ArgumentException](#)

The **identity** does not refer to a [PlcType](#) that addresses data of the raw type [LWord](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

The number of **values** is outside of the bounds defined by [MinLength](#) and [MaxLength](#).

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# WriteLReal(PlcIdentity, IEnumerable<Double>)

Writes the [Double](#) array (in PLC an ARRAY OF LWORD) **values** to the data area associated with the **identity** specified.

## C#

```
[CLSCompliant(false)]
public void WriteLReal(PlcIdentity identity, IEnumerable<double> values)
```

## Parameters

**identity** [PlcIdentity](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

**values** [IEnumerable<Double>](#)

The values to write to the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` does not refer to a `PlcType` that addresses data of the raw type `LWord`.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of `values` is outside of the bounds defined by `MinLength` and `MaxLength`.

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteLReal(PlcLRealArrayType, Double[])

Writes the `Double` array (in PLC an ARRAY OF LWORD) `values` to the data area associated with the `type` specified.

## C#

```
[CLSCompliant(false)]
public void WriteLReal(PlcLRealArrayType type, params double[] values)
```

## Parameters

### type PlcLRealArrayType

The `PlcLRealArrayType` which addresses the data area to write.

### values Double[]

The values to write to the data area identified by the `type` specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# WriteLReal(PlcLRealArrayType, IEnumerable<Double>)

Writes the **Double** array (in PLC an ARRAY OF LWORD) **values** to the data area associated with the **type** specified.

## C#

```
[CLSCompliant(false)]  
public void WriteLReal(PlcLRealArrayType type, IEnumerable<double> values)
```

## Parameters

**type** **PlcLRealArrayType**

The **PlcLRealArrayType** which addresses the data area to write.

**values** **IEnumerable<Double>**

The values to write to the data area identified by the **type** specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteLReal(PlcLRealType, Double)

Writes the **Double** (in PLC a LWORD) **value** using the **type** specified.

## C#

```
[CLSCompliant(false)]  
public void WriteLReal(PlcLRealType type, double value)
```

## Parameters

**type** **PlcLRealType**

The **PlcLRealType** which addresses the data area to write.

value Double

The value to write to the data area identified by the type specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in Opened, Connected or Disconnected state and is therefore not ready to use or the connection is in Faulted state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteLReal(String, Double)

Writes the Double (in PLC a LWORD) value to the data area associated with the identity specified.

## C#

```
[CLSCompliant(false)]
public void WriteLReal(string identity, double value)
```

## Parameters

identity String

The PlcIdentity that can be either a PlcAddress which addresses the data area to write. Or a PlcName representing the symbolic name of the PlcType (to write) associated with the Device of this connection.

value Double

The value to write to the data area identified by the identity specified.

## Exceptions

### ArgumentException

The identity is an empty string or does not refer to a PlcType that addresses data of the raw type LWord.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in Opened, Connected or Disconnected state and is therefore not ready to use or the connection is in Faulted state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteLReal(String, Double[])

Writes the [Double](#) array (in PLC an ARRAY OF LWORD) [values](#) to the data area associated with the [identity](#) specified.

**C#**

```
[CLSCompliant(false)]
public void WriteLReal(string identity, params double[] values)
```

### Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

[values](#) [Double\[\]](#)

The values to write to the data area identified by the [identity](#) specified.

### Exceptions

[ArgumentException](#)

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [LWord](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

The number of [values](#) is outside of the bounds defined by [MinLength](#) and [MaxLength](#).

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## WriteLReal(String, IEnumerable<Double>)

Writes the [Double](#) array (in PLC an ARRAY OF LWORD) [values](#) to the data area associated with the [identity](#) specified.

**C#**

```
[CLSCompliant(false)]
public void WriteLReal(string identity, IEnumerable<double> values)
```

## Parameters

**identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** IEnumerable<Double>

The values to write to the data area identified by the **identity** specified.

## Exceptions

**ArgumentException**

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **LWord**.

**ArgumentNullException**

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

**ArgumentOutOfRangeException**

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# WriteLRealCore(PlcType, Double[])

Writes the data area described by the **type** specified.

**C#**

```
protected virtual void WriteLRealCore(PlcType type, params double[] values)
```

## Parameters

**type** PlcType

The **PlcLRealType** or **PlcLRealArrayType** to write.

**values** Double[]

The array of **Double** data to write.

## WriteNodeCore(PlcDataNode)

Writes the data area described by the **node** specified.

**C#**

```
protected virtual void WriteNodeCore(PlcDataNode node)
```

### Parameters

**node** PlcDataNode

The **PlcDataNode** to write.

## WriteNodesCore(IEnumerable<PlcDataNode>)

When implemented in a derived class, writes the data areas described by the **nodes** specified.

**C#**

```
protected abstract void WriteNodesCore(IEnumerable<PlcDataNode> nodes)
```

### Parameters

**nodes** IEnumerable<PlcDataNode>

The **PlcDataNode**'s to write.

## WriteNodesCore(PlcDataNode[])

Writes the data areas described by the **nodes** specified.

**C#**

```
protected void WriteNodesCore(params PlcDataNode[] nodes)
```

### Parameters

**nodes** PlcDataNode[]

The **PlcDataNode**'s to write.

## WriteObject<T>(T)

Writes the **PlcObject** specified by **T** using the **instance** passed.

**C#**

```
public void WriteObject<T>(T instance)
    where T : PlcObject
```



## Parameters

**instance** T

The instance of T to write.

## Exceptions

**ArgumentNullException**

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

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# WriteReal(PlcIdentity, IEnumerable<Single>)

Writes the **Single** array (in PLC an ARRAY OF DWORD) **values** to the data area associated with the **identity** specified.

**C#**

```
[CLSCompliant(false)]
public void WriteReal(PlcIdentity identity, IEnumerable<float> values)
```

## Parameters

**identity** PlcIdentity

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** IEnumerable<Single>

The values to write to the data area identified by the **identity** specified.

## Exceptions

**ArgumentException**

The **identity** does not refer to a **PlcType** that addresses data of the raw type **DWord**.

**ArgumentNullException**

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

**ArgumentOutOfRangeException**

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteReal(PlcIdentity, Single)

Writes the [Single](#) (in PLC a DWORD) [value](#) to the data area associated with the [identity](#) specified.

### C#

```
[CLSCompliant(false)]
public void WriteReal(PlcIdentity identity, float value)
```

### Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

[value](#) [Single](#)

The value to write to the data area identified by the [identity](#) specified.

### Exceptions

#### ArgumentException

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

#### ArgumentNullException

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

#### InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## WriteReal(PlcIdentity, Single[])

Writes the [Single](#) array (in PLC an ARRAY OF DWORD) [values](#) to the data area associated with the [identity](#) specified.

### C#

```
[CLSCompliant(false)]
public void WriteReal(PlcIdentity identity, params float[] values)
```

## Parameters

**identity** [PlcIdentity](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

**values** [Single\[\]](#)

The values to write to the data area identified by the **identity** specified.

## Exceptions

[ArgumentException](#)

The **identity** does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

The number of **values** is outside of the bounds defined by [MinLength](#) and [MaxLength](#).

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# WriteReal(PlcRealArrayType, IEnumerable<Single>)

Writes the [Single](#) array (in PLC an ARRAY OF DWORD) **values** to the data area associated with the **type** specified.

## C#

```
[CLSCompliant(false)]
public void WriteReal(PlcRealArrayType type, IEnumerable<float> values)
```

## Parameters

**type** [PlcRealArrayType](#)

The [PlcRealArrayType](#) which addresses the data area to write.

**values** [IEnumerable<Single>](#)

The values to write to the data area identified by the **type** specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteReal(PlcRealArrayType, Single[])

Writes the **Single** array (in PLC an ARRAY OF DWORD) **values** to the data area associated with the **type** specified.

## C#

```
[CLSCompliant(false)]
public void WriteReal(PlcRealArrayType type, params float[] values)
```

## Parameters

**type** **PlcRealArrayType**

The **PlcRealArrayType** which addresses the data area to write.

**values** **Single[]**

The values to write to the data area identified by the **type** specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteReal(PlcRealType, Single)

Writes the **Single** (in PLC a DWORD) **value** using the **type** specified.

## C#

```
[CLSCompliant(false)]  
public void WriteReal(PlcRealType type, float value)
```

## Parameters

**type** **PlcRealType**

The **PlcRealType** which addresses the data area to write.

**value** **Single**

The value to write to the data area identified by the **type** specified.

## Exceptions

**ArgumentNullException**

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

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# WriteReal(String, IEnumerable<Single>)

Writes the **Single** array (in PLC an ARRAY OF DWORD) **values** to the data area associated with the **identity** specified.

## C#

```
[CLSCompliant(false)]  
public void WriteReal(string identity, IEnumerable<float> values)
```

## Parameters

**identity** **String**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** **IEnumerable<Single>**

The values to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **DWord**.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteReal(String, Single)

Writes the **Single** (in PLC a DWORD) **value** to the data area associated with the **identity** specified.

### C#

```
[CLSCompliant(false)]
public void WriteReal(string identity, float value)
```

## Parameters

### **identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

### **value** Single

The value to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **DWord**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# WriteReal(String, Single[])

Writes the [Single](#) array (in PLC an ARRAY OF DWORD) [values](#) to the data area associated with the [identity](#) specified.

## C#

```
[CLSCompliant(false)]
public void WriteReal(string identity, params float[] values)
```

## Parameters

[identity](#) [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

[values](#) [Single\[\]](#)

The values to write to the data area identified by the [identity](#) specified.

## Exceptions

[ArgumentException](#)

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

The number of [values](#) is outside of the bounds defined by [MinLength](#) and [MaxLength](#).

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# WriteRealCore(PlcType, Single[])

Writes the data area described by the [type](#) specified.

## C#

```
protected virtual void WriteRealCore(PlcType type, params float[] values)
```

## Parameters

**type** `PlcType`

The `PlcRealType` or `PlcRealArrayType` to write.

**values** `Single[]`

The array of `Single` data to write.

# WriteS5Time(PlcIdentity, TimeSpan)

Writes the `TimeSpan` (in PLC a S5TIME) **value** to the data area associated with the **identity** specified.

## C#

```
public void WriteS5Time(PlcIdentity identity, TimeSpan value)
```

## Parameters

**identity** `PlcIdentity`

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to write. Or a `PlcName` representing the symbolic name of the `PlcType` (to write) associated with the `Device` of this connection.

**value** `TimeSpan`

The value to write to the data area identified by the **identity** specified.

## Exceptions

`ArgumentException`

The **identity** does not refer to a `PlcType` that addresses data of the raw type `Word`.

`ArgumentNullException`

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

`InvalidOperationException`

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

`ObjectDisposedException`

The connection has been disposed of.

# WriteS5Time(PlcS5TimeType, TimeSpan)

Writes the `TimeSpan` (in PLC a S5TIME) **value** using the **type** specified.

## C#

```
public void WriteS5Time(PlcS5TimeType type, TimeSpan value)
```



## Parameters

**type** [PlcS5TimeType](#)

The [PlcS5TimeType](#) which addresses the data area to write.

**value** [TimeSpan](#)

The value to write to the data area identified by the **type** specified.

## Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# WriteS5Time(String, TimeSpan)

Writes the [TimeSpan](#) (in PLC a S5TIME) **value** to the data area associated with the **identity** specified.

## C#

```
public void WriteS5Time(string identity, TimeSpan value)
```

## Parameters

**identity** [String](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

**value** [TimeSpan](#)

The value to write to the data area identified by the **identity** specified.

## Exceptions

[ArgumentException](#)

The **identity** is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [Word](#).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

#### [ObjectDisposedException](#)

The connection has been disposed of.

## WriteS5TimeCore(PlcType, TimeSpan)

Writes the data area described by the [type](#) specified.

#### C#

```
protected virtual void WriteS5TimeCore(PlcType type, TimeSpan value)
```

#### Parameters

[type](#) [PlcType](#)

The [PlcS5TimeType](#) to write.

[value](#) [TimeSpan](#)

The [TimeSpan](#) data to write.

## WriteString(PlcIdentity, String)

Writes the [String](#) (in PLC a STRING) [value](#) to the data area associated with the [identity](#) specified.

#### C#

```
public void WriteString(PlcIdentity identity, string value)
```

#### Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

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

The value to write to the data area identified by the [identity](#) specified.

#### Exceptions

##### [ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [Byte](#).

##### [ArgumentNullException](#)

The [identity](#) or [value](#) is a null reference (Nothing in Visual Basic).

### InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteString(PlcStringType, String)

Writes the [String](#) (in PLC a STRING) [value](#) using the [type](#) specified.

### C#

```
public void WriteString(PlcStringType type, string value)
```

### Parameters

[type](#) [PlcStringType](#)

The [PlcStringType](#) which addresses the data area to write.

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

The value to write to the data area identified by the [type](#) specified.

### Exceptions

#### ArgumentNullException

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

#### InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## WriteString(String, String)

Writes the [TimeSpan](#) (in PLC a STRING) [value](#) to the data area associated with the [identity](#) specified.

### C#

```
public void WriteString(string identity, string value)
```

### Parameters

[identity](#) [String](#)

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**value** String

The value to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Byte**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteStringCore(PlcType, String)

Writes the data area described by the **type** specified.

### C#

```
protected virtual void WriteStringCore(PlcType type, string value)
```

## Parameters

**type** PlcType

The **PlcStringType** to write.

**value** String

The **String** data to write.

## WriteTime(PlcIdentity, TimeSpan)

Writes the **TimeSpan** (in PLC a TIME) **value** to the data area associated with the **identity** specified.

### C#

```
public void WriteTime(PlcIdentity identity, TimeSpan value)
```

## Parameters

## identity PlcIdentity

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

## value TimeSpan

The value to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** does not refer to a **PlcType** that addresses data of the raw type **DWord**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteTime(PlcTimeType, TimeSpan)

Writes the **TimeSpan** (in PLC a TIME) **value** using the **type** specified.

## C#

```
public void WriteTime(PlcTimeType type, TimeSpan value)
```

## Parameters

### type PlcTimeType

The **PlcTimeType** which addresses the data area to write.

### value TimeSpan

The value to write to the data area identified by the **type** specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the

connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteTime(String, TimeSpan)

Writes the **TimeSpan** (in PLC a TIME) **value** to the data area associated with the **identity** specified.

### C#

```
public void WriteTime(string identity, TimeSpan value)
```

### Parameters

#### identity String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

#### value TimeSpan

The value to write to the data area identified by the **identity** specified.

### Exceptions

#### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **DWord**.

#### ArgumentNullException

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

#### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## WriteTimeCore(PlcType, TimeSpan)

Writes the data area described by the **type** specified.

### C#

```
protected virtual void WriteTimeCore(PlcType type, TimeSpan value)
```

### Parameters

**type** `PlcType`

The `PlcTimeType` to write.

**value** `TimeSpan`

The `TimeSpan` data to write.

## WriteTimeOfDay(PlcIdentity, TimeSpan)

Writes the `TimeSpan` (in PLC a TIME\_OF\_DAY) **value** to the data area associated with the **identity** specified.

**C#**

```
public void WriteTimeOfDay(PlcIdentity identity, TimeSpan value)
```

### Parameters

**identity** `PlcIdentity`

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to write. Or a `PlcName` representing the symbolic name of the `PlcType` (to write) associated with the `Device` of this connection.

**value** `TimeSpan`

The value to write to the data area identified by the **identity** specified.

### Exceptions

`ArgumentException`

The **identity** does not refer to a `PlcType` that addresses data of the raw type `DWord`.

`ArgumentNullException`

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

`InvalidOperationException`

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

`ObjectDisposedException`

The connection has been disposed of.

## WriteTimeOfDay(PlcTimeOfDayType, TimeSpan)

Writes the `TimeSpan` (in PLC a TIME\_OF\_DAY) **value** using the **type** specified.

**C#**

```
public void WriteTimeOfDay(PlcTimeOfDayType type, TimeSpan value)
```

## Parameters

**type** `PlcTimeOfDayType`

The `PlcTimeOfDayType` which addresses the data area to write.

**value** `TimeSpan`

The value to write to the data area identified by the `type` specified.

## Exceptions

`ArgumentNullException`

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

`InvalidOperationException`

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

`ObjectDisposedException`

The connection has been disposed of.

# WriteTimeOfDay(String, TimeSpan)

Writes the `TimeSpan` (in PLC a `TIME_OF_DAY`) `value` to the data area associated with the `identity` specified.

## C#

```
public void WriteTimeOfDay(string identity, TimeSpan value)
```

## Parameters

**identity** `String`

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to write. Or a `PlcName` representing the symbolic name of the `PlcType` (to write) associated with the `Device` of this connection.

**value** `TimeSpan`

The value to write to the data area identified by the `identity` specified.

## Exceptions

`ArgumentException`

The `identity` is an empty string or does not refer to a `PlcType` that addresses data of the raw type `DWord`.

`ArgumentNullException`

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



### InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteTimeOfDayCore(PlcType, TimeSpan)

Writes the data area described by the [type](#) specified.

### C#

```
protected virtual void WriteTimeOfDayCore(PlcType type, TimeSpan value)
```

### Parameters

[type](#) [PlcType](#)

The [PlcTimeOfDayType](#) to write.

[value](#) [TimeSpan](#)

The [TimeSpan](#) data to write.

## WriteUInt16(PlcIdentity, IEnumerable<UInt16>)

Writes the [UInt16](#) array (in PLC an ARRAY OF WORD) [values](#) to the data area associated with the [identity](#) specified.

### C#

```
[CLSCompliant(false)]  
public void WriteUInt16(PlcIdentity identity, IEnumerable<ushort> values)
```

### Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

[values](#) [IEnumerable<UInt16>](#)

The values to write to the data area identified by the [identity](#) specified.

### Exceptions

#### ArgumentException

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [Word](#).

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteUInt16(PlcIdentity, UInt16)

Writes the **UInt16** (in PLC a WORD) **value** to the data area associated with the **identity** specified.

### C#

```
[CLSCompliant(false)]
public void WriteUInt16(PlcIdentity identity, ushort value)
```

### Parameters

**identity** **PlcIdentity**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**value** **UInt16**

The value to write to the data area identified by the **identity** specified.

### Exceptions

#### ArgumentException

The **identity** does not refer to a **PlcType** that addresses data of the raw type **Word**.

#### ArgumentNullException

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

#### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

# WriteUInt16(PlcIdentity, UInt16[])

Writes the **UInt16** array (in PLC an ARRAY OF WORD) **values** to the data area associated with the **identity** specified.

**C#**

```
[CLSCompliant(false)]  
public void WriteUInt16(PlcIdentity identity, params ushort[] values)
```

## Parameters

**identity** **PlcIdentity**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** **UInt16[]**

The values to write to the data area identified by the **identity** specified.

## Exceptions

**ArgumentException**

The **identity** does not refer to a **PlcType** that addresses data of the raw type **Word**.

**ArgumentNullException**

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

**ArgumentOutOfRangeException**

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# WriteUInt16(PlcUInt16ArrayType, IEnumerable<UInt16>)

Writes the **UInt16** array (in PLC an ARRAY OF WORD) **values** to the data area associated with the **type** specified.

**C#**

```
[CLSCompliant(false)]  
public void WriteUInt16(PlcUInt16ArrayType type, IEnumerable<ushort> values)
```

## Parameters

**type** `PlcUInt16ArrayType`

The `PlcUInt16ArrayType` which addresses the data area to write.

**values** `IEnumerable<UInt16>`

The values to write to the data area identified by the **type** specified.

## Exceptions

`ArgumentNullException`

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

`InvalidOperationException`

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

`ObjectDisposedException`

The connection has been disposed of.

# WriteUInt16(PlcUInt16ArrayType, UInt16[])

Writes the `UInt16` array (in PLC an ARRAY OF WORD) **values** to the data area associated with the **type** specified.

## C#

```
[CLSCompliant(false)]
public void WriteUInt16(PlcUInt16ArrayType type, params ushort[] values)
```

## Parameters

**type** `PlcUInt16ArrayType`

The `PlcUInt16ArrayType` which addresses the data area to write.

**values** `UInt16[]`

The values to write to the data area identified by the **type** specified.

## Exceptions

`ArgumentNullException`

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

`InvalidOperationException`

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# WriteUInt16(PlcUInt16Type, UInt16)

Writes the **UInt16** (in PLC a WORD) **value** using the **type** specified.

## C#

```
[CLSCompliant(false)]  
public void WriteUInt16(PlcUInt16Type type, ushort value)
```

## Parameters

**type** **PlcUInt16Type**

The **PlcUInt16Type** which addresses the data area to write.

**value** **UInt16**

The value to write to the data area identified by the **type** specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteUInt16(String, IEnumerable<UInt16>)

Writes the **UInt16** array (in PLC an ARRAY OF WORD) **values** to the data area associated with the **identity** specified.

## C#

```
[CLSCompliant(false)]  
public void WriteUInt16(string identity, IEnumerable<ushort> values)
```

## Parameters

**identity** **String**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName**

representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** **IEnumerable<UInt16>**

The values to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Word**.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteUInt16(String, UInt16)

Writes the **UInt16** (in PLC a WORD) **value** to the data area associated with the **identity** specified.

## C#

```
[CLSCompliant(false)]
public void WriteUInt16(string identity, ushort value)
```

## Parameters

**identity** **String**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**value** **UInt16**

The value to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Word**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteUInt16(String, UInt16[])

Writes the **UInt16** array (in PLC an ARRAY OF WORD) **values** to the data area associated with the **identity** specified.

### C#

```
[CLSCompliant(false)]
public void WriteUInt16(string identity, params ushort[] values)
```

### Parameters

**identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** UInt16[]

The values to write to the data area identified by the **identity** specified.

### Exceptions

#### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **Word**.

#### ArgumentNullException

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

#### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

#### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

# WriteUInt16Core(PlcType, UInt16[])

Writes the data area described by the **type** specified.

**C#**

```
[CLSCompliant(false)]
protected virtual void WriteUInt16Core(PlcType type, params ushort[] values)
```

## Parameters

**type** PlcType

The PlcUInt16Type or PlcUInt16ArrayType to write.

**values** UInt16[]

The array of UInt16 data to write.

# WriteUInt32(PlcIdentity, IEnumerable<UInt32>)

Writes the UInt32 array (in PLC an ARRAY OF DWORD) **values** to the data area associated with the **identity** specified.

**C#**

```
[CLSCompliant(false)]
public void WriteUInt32(PlcIdentity identity, IEnumerable<uint> values)
```

## Parameters

**identity** PlcIdentity

The PlcIdentity that can be either a PlcAddress which addresses the data area to write. Or a PlcName representing the symbolic name of the PlcType (to write) associated with the Device of this connection.

**values** IEnumerable<UInt32>

The values to write to the data area identified by the **identity** specified.

## Exceptions

ArgumentException

The **identity** does not refer to a PlcType that addresses data of the raw type DWord.

ArgumentNullException

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

ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by MinLength and MaxLength.

InvalidOperationException



The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

#### [ObjectDisposedException](#)

The connection has been disposed of.

## WriteUInt32(PlcIdentity, UInt32)

Writes the [UInt32](#) (in PLC a DWORD) [value](#) to the data area associated with the [identity](#) specified.

### C#

```
[CLSCompliant(false)]  
public void WriteUInt32(PlcIdentity identity, uint value)
```

### Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

[value](#) [UInt32](#)

The value to write to the data area identified by the [identity](#) specified.

### Exceptions

#### [ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

#### [ArgumentNullException](#)

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

#### [InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

#### [ObjectDisposedException](#)

The connection has been disposed of.

## WriteUInt32(PlcIdentity, UInt32[])

Writes the [UInt32](#) array (in PLC an ARRAY OF DWORD) [values](#) to the data area associated with the [identity](#) specified.

### C#

```
[CLSCompliant(false)]  
public void WriteUInt32(PlcIdentity identity, params uint[] values)
```

## Parameters

**identity** [PlcIdentity](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

**values** [UInt32\[\]](#)

The values to write to the data area identified by the **identity** specified.

## Exceptions

[ArgumentException](#)

The **identity** does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

The number of **values** is outside of the bounds defined by [MinLength](#) and [MaxLength](#).

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# WriteUInt32(PlcUInt32ArrayType, IEnumerable<UInt32>)

Writes the [UInt32](#) array (in PLC an ARRAY OF DWORD) **values** to the data area associated with the **type** specified.

## C#

```
[CLSCompliant(false)]  
public void WriteUInt32(PlcUInt32ArrayType type, IEnumerable<uint> values)
```

## Parameters

**type** [PlcUInt32ArrayType](#)

The [PlcUInt32ArrayType](#) which addresses the data area to write.

**values** [IEnumerable<UInt32>](#)

The values to write to the data area identified by the **type** specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteUInt32(PlcUInt32ArrayType, UInt32[])

Writes the **UInt32** array (in PLC an ARRAY OF DWORD) **values** to the data area associated with the **type** specified.

### C#

```
[CLSCompliant(false)]
public void WriteUInt32(PlcUInt32ArrayType type, params uint[] values)
```

## Parameters

**type** **PlcUInt32ArrayType**

The **PlcUInt32ArrayType** which addresses the data area to write.

**values** **UInt32[]**

The values to write to the data area identified by the **type** specified.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteUInt32(PlcUInt32Type, UInt32)

Writes the **UInt32** (in PLC a DWORD) **value** using the **type** specified.

## C#

```
[CLSCompliant(false)]
public void WriteUInt32(PlcUInt32Type type, uint value)
```

## Parameters

**type** **PlcUInt32Type**

The **PlcUInt32Type** which addresses the data area to write.

**value** **UInt32**

The value to write to the data area identified by the **type** specified.

## Exceptions

**ArgumentNullException**

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

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# WriteUInt32(String, IEnumerable<UInt32>)

Writes the **UInt32** array (in PLC an ARRAY OF DWORD) **values** to the data area associated with the **identity** specified.

## C#

```
[CLSCompliant(false)]
public void WriteUInt32(string identity, IEnumerable<uint> values)
```

## Parameters

**identity** **String**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** **IEnumerable<UInt32>**

The values to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **DWord**.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteUInt32(String, UInt32)

Writes the **UInt32** (in PLC a DWORD) **value** to the data area associated with the **identity** specified.

### C#

```
[CLSCompliant(false)]
public void WriteUInt32(string identity, uint value)
```

## Parameters

### **identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

### **value** UInt32

The value to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **DWord**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

## ObjectDisposedException

The connection has been disposed of.

# WriteUInt32(String, UInt32[])

Writes the [UInt32](#) array (in PLC an ARRAY OF DWORD) [values](#) to the data area associated with the [identity](#) specified.

## C#

```
[CLSCompliant(false)]
public void WriteUInt32(string identity, params uint[] values)
```

## Parameters

[identity](#) String

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

[values](#) UInt32[]

The values to write to the data area identified by the [identity](#) specified.

## Exceptions

[ArgumentException](#)

The [identity](#) is an empty string or does not refer to a [PlcType](#) that addresses data of the raw type [DWord](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

The number of [values](#) is outside of the bounds defined by [MinLength](#) and [MaxLength](#).

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# WriteUInt32Core(PlcType, UInt32[])

Writes the data area described by the [type](#) specified.

## C#

```
[CLSCompliant(false)]
protected virtual void WriteUInt32Core(PlcType type, params uint[] values)
```

## Parameters

**type** [PlcType](#)

The [PlcUInt32Type](#) or [PlcUInt32ArrayType](#) to write.

**values** [UInt32\[\]](#)

The array of [UInt32](#) data to write.

# WriteUInt64(PlcIdentity, IEnumerable<UInt64>)

Writes the [UInt64](#) array (in PLC an ARRAY OF LWORD) **values** to the data area associated with the **identity** specified.

## C#

```
[CLSCompliant(false)]
public void WriteUInt64(PlcIdentity identity, IEnumerable<ulong> values)
```

## Parameters

**identity** [PlcIdentity](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

**values** [IEnumerable<UInt64>](#)

The values to write to the data area identified by the **identity** specified.

## Exceptions

[ArgumentException](#)

The **identity** does not refer to a [PlcType](#) that addresses data of the raw type [LWord](#).

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

The number of **values** is outside of the bounds defined by [MinLength](#) and [MaxLength](#).

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## WriteUInt64(PlcIdentity, UInt64)

Writes the [UInt64](#) (in PLC a LWORD) [value](#) to the data area associated with the [identity](#) specified.

### C#

```
[CLSCompliant(false)]  
public void WriteUInt64(PlcIdentity identity, ulong value)
```

### Parameters

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

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

[value](#) [UInt64](#)

The value to write to the data area identified by the [identity](#) specified.

### Exceptions

[ArgumentException](#)

The [identity](#) does not refer to a [PlcType](#) that addresses data of the raw type [LWord](#).

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## WriteUInt64(PlcIdentity, UInt64[])

Writes the [UInt64](#) array (in PLC an ARRAY OF LWORD) [values](#) to the data area associated with the [identity](#) specified.

### C#

```
[CLSCompliant(false)]  
public void WriteUInt64(PlcIdentity identity, params ulong[] values)
```

### Parameters



## identity PlcIdentity

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

## values UInt64[]

The values to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** does not refer to a **PlcType** that addresses data of the raw type **LWord**.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteUInt64(PlcUInt64ArrayType, IEnumerable<UInt64>)

Writes the **UInt64** array (in PLC an ARRAY OF LWORD) **values** to the data area associated with the **type** specified.

## C#

```
[CLSCompliant(false)]
public void WriteUInt64(PlcUInt64ArrayType type, IEnumerable<ulong> values)
```

## Parameters

### type PlcUInt64ArrayType

The **PlcUInt64ArrayType** which addresses the data area to write.

### values IEnumerable<UInt64>

The values to write to the data area identified by the **type** specified.

## Exceptions

### ArgumentNullException

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

#### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## WriteUInt64(PlcUInt64ArrayType, UInt64[])

Writes the **UInt64** array (in PLC an ARRAY OF LWORD) **values** to the data area associated with the **type** specified.

### C#

```
[CLSCompliant(false)]
public void WriteUInt64(PlcUInt64ArrayType type, params ulong[] values)
```

### Parameters

**type** **PlcUInt64ArrayType**

The **PlcUInt64ArrayType** which addresses the data area to write.

**values** **UInt64[]**

The values to write to the data area identified by the **type** specified.

### Exceptions

#### ArgumentNullException

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

#### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## WriteUInt64(PlcUInt64Type, UInt64)

Writes the **UInt64** (in PLC a LWORD) **value** using the **type** specified.

### C#

```
[CLSCompliant(false)]
public void WriteUInt64(PlcUInt64Type type, ulong value)
```

## Parameters

**type** `PlcUInt64Type`

The `PlcUInt64Type` which addresses the data area to write.

**value** `UInt64`

The value to write to the data area identified by the `type` specified.

## Exceptions

`ArgumentNullException`

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

`InvalidOperationException`

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

`ObjectDisposedException`

The connection has been disposed of.

# WriteUInt64(String, IEnumerable<UInt64>)

Writes the `UInt64` array (in PLC an ARRAY OF LWORD) `values` to the data area associated with the `identity` specified.

## C#

```
[CLSCompliant(false)]
public void WriteUInt64(string identity, IEnumerable<ulong> values)
```

## Parameters

**identity** `String`

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to write. Or a `PlcName` representing the symbolic name of the `PlcType` (to write) associated with the `Device` of this connection.

**values** `IEnumerable<UInt64>`

The values to write to the data area identified by the `identity` specified.

## Exceptions

`ArgumentException`

The `identity` is an empty string or does not refer to a `PlcType` that addresses data of the raw type `LWord`.

`ArgumentNullException`

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

### ArgumentOutOfRangeException

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteUInt64(String, UInt64)

Writes the **UInt64** (in PLC a LWORD) **value** to the data area associated with the **identity** specified.

### C#

```
[CLSCompliant(false)]
public void WriteUInt64(string identity, ulong value)
```

### Parameters

#### **identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

#### **value** UInt64

The value to write to the data area identified by the **identity** specified.

### Exceptions

#### ArgumentException

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **LWord**.

#### ArgumentNullException

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

#### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## WriteUInt64(String, UInt64[])

Writes the **UInt64** array (in PLC an ARRAY OF LWORD) **values** to the data area associated with the

**identity** specified.

**C#**

```
[CLSCompliant(false)]
public void WriteUInt64(string identity, params ulong[] values)
```

## Parameters

**identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** UInt64[]

The values to write to the data area identified by the **identity** specified.

## Exceptions

**ArgumentException**

The **identity** is an empty string or does not refer to a **PlcType** that addresses data of the raw type **LWord**.

**ArgumentNullException**

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

**ArgumentOutOfRangeException**

The number of **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# WriteUInt64Core(PlcType, UInt64[])

Writes the data area described by the **type** specified.

**C#**

```
[CLSCompliant(false)]
protected virtual void WriteUInt64Core(PlcType type, params ulong[] values)
```

## Parameters

**type** PlcType

The **PlcUInt64Type** or **PlcUInt64ArrayType** to write.

values UInt64[]

The array of UInt64 data to write.

## WriteValue<T>(IPlcValue<T>)

Writes the specified value.

**C#**

```
public void WriteValue<T>(IPlcValue<T> value)
```

### Parameters

value IPlcValue<T>

The IPlcValue`1 to write.

### Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

## WriteValue<T>(PlcArrayType, IEnumerable<T>)

Writes the T using the type specified.

**C#**

```
public void WriteValue<T>(PlcArrayType type, IEnumerable<T> values)
```

### Parameters

type PlcArrayType

The PlcArrayType which addresses the data area to write.

values IEnumerable<T>

The values to write to the data area identified by the type specified.

## Exceptions

### ArgumentException

The `type` specified can not be used to access the framework type specified by `T`.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteValue<T>(PlcArrayType, T[])

Writes the `T` using the `type` specified.

### C#

```
public void WriteValue<T>(PlcArrayType type, params T[] values)
```

### Parameters

`type` `PlcArrayType`

The `PlcArrayType` which addresses the data area to write.

`values` `T`

The values to write to the data area identified by the `type` specified.

## Exceptions

### ArgumentException

The `type` specified can not be used to access the framework type specified by `T`.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteValue<T>(PlcIdentity, IEnumerable<T>)

Writes an array of the **T** which is associated with the **identity** specified.

**C#**

```
public void WriteValue<T>(PlcIdentity identity, IEnumerable<T> values)
```

## Parameters

**identity** [PlcIdentity](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

**values** [IEnumerable<T>](#)

The array to write to the data area identified by the **identity** specified.

## Exceptions

[ArgumentException](#)

The metadata specified could not be resolved to a valid [PlcArrayType](#) or the **identity** does not refer to a [PlcType](#) that can be used to access the framework type specified by **T**.

[ArgumentNullException](#)

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

[ArgumentOutOfRangeException](#)

The number of items in **values** is outside of the bounds defined by [MinLength](#) and [MaxLength](#).

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# WriteValue<T>(PlcIdentity, T)

Writes the **T** which is associated with the **identity** specified.

**C#**

```
public void WriteValue<T>(PlcIdentity identity, T value)
```

## Parameters

**identity** [PlcIdentity](#)



The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

value **T**

The value to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The metadata specified could not be resolved to a valid **PlcValueType** or the **identity** does not refer to a **PlcType** that can be used to access the framework type specified by **T**.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

# WriteValue<T>(PlcIdentity, T[])

Writes an array of the **T** which is associated with the **identity** specified.

**C#**

```
public void WriteValue<T>(PlcIdentity identity, params T[] values)
```

## Parameters

**identity** **PlcIdentity**

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

values **T**

The array to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The metadata specified could not be resolved to a valid **PlcArrayType** or the **identity** does not refer to a **PlcType** that can be used to access the framework type specified by **T**.

### ArgumentNullException

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

#### ArgumentOutOfRangeException

The number of items in **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

#### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## WriteValue<T>(PlcValueType, T)

Writes the **T** using the **type** specified.

### C#

```
public void WriteValue<T>(PlcValueType type, T value)
```

### Parameters

**type** PlcValueType

The **PlcValueType** which addresses the data area to write.

**value** T

The value to write to the data area identified by the **type** specified.

### Exceptions

#### ArgumentException

The **type** specified can not be used to access the framework type specified by **T**.

#### ArgumentNullException

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

#### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## WriteValue<T>(String, IEnumerable<T>)

Writes an array of the **T** which is associated with the **identity** specified.

## C#

```
public void WriteValue<T>(string identity, IEnumerable<T> values)
```

### Parameters

**identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** IEnumerable<T>

The array to write to the data area identified by the **identity** specified.

### Exceptions

**ArgumentException**

The **identity** is an empty string or the metadata specified could not be resolved to a valid **PlcArrayType** or the **identity** does not refer to a **PlcType** that can be used to access the framework type specified by **T**.

**ArgumentNullException**

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

**ArgumentOutOfRangeException**

The number of items in **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

## WriteValue<T>(String, T)

Writes the **T** which is associated with the **identity** specified.

## C#

```
public void WriteValue<T>(string identity, T value)
```

### Parameters

**identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**value** T

The value to write to the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` is an empty string or the metadata specified could not be resolved to a valid `PlcValueType` or the `identity` does not refer to a `PlcType` that can be used to access the framework type specified by `T`.

### ArgumentNullException

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

### InvalidOperationException

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteValue<T>(String, T[])

Writes an array of the `T` which is associated with the `identity` specified.

### C#

```
public void WriteValue<T>(string identity, params T[] values)
```

## Parameters

### `identity` String

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to write. Or a `PlcName` representing the symbolic name of the `PlcType` (to write) associated with the `Device` of this connection.

### `values` T

The array to write to the data area identified by the `identity` specified.

## Exceptions

### ArgumentException

The `identity` is an empty string or the metadata specified could not be resolved to a valid `PlcArrayType` or the `identity` does not refer to a `PlcType` that can be used to access the framework type specified by `T`.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of items in **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteValueCore<T>(PlcArrayType, IEnumerable<T>)

Writes the data area described by the **type** specified.

### C#

```
protected virtual void WriteValueCore<T>(PlcArrayType type, IEnumerable<T> values)
```

### Parameters

**type** PlcArrayType

The **PlcArrayType** to write.

**values** IEnumerable<T>

The values data to write.

## WriteValueCore<T>(PlcArrayType, T[])

Writes the data area described by the **type** specified.

### C#

```
protected virtual void WriteValueCore<T>(PlcArrayType type, params T[] values)
```

### Parameters

**type** PlcArrayType

The **PlcArrayType** to write.

**values** T

The values data to write.

## WriteValueCore<T>(PlcValueType, T)

Writes the data area described by the **type** specified.

### C#

```
protected virtual void WriteValueCore<T>(PlcValueType type, T value)
```

## Parameters

**type** `PlcValueType`

The `PlcValueType` to write.

**value** `T`

The value data to write.

# WriteValues(IEnumerable<IPlcValue>)

Writes the values specified.

## C#

```
public void WriteValues(IEnumerable<IPlcValue> values)
```

## Parameters

**values** `IEnumerable<IPlcValue>`

The `IPlcValue` objects to write.

## Exceptions

[ArgumentNullException](#)

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

[InvalidOperationException](#)

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

[ObjectDisposedException](#)

The connection has been disposed of.

# WriteValues(IPlcValue[])

Writes the values specified.

## C#

```
public void WriteValues(params IPlcValue[] values)
```

## Parameters

**values** `IPlcValue[]`

The `IPlcValue` objects to write.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteValues<T>(IEnumerable<IPlcValue<T>>)

Writes the values specified.

### C#

```
public void WriteValues<T>(IEnumerable<IPlcValue<T>> values)
```

## Parameters

**values** **IEnumerable<IPlcValue>**

The **IPlcValue**'1 objects to write.

## Exceptions

### ArgumentNullException

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

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteValues<T>(IPlcValue<T>[])

Writes the values specified.

### C#

```
public void WriteValues<T>(params IPlcValue<T>[] values)
```

## Parameters

**values** `IPlcValue<T>[]`

The `IPlcValue` objects to write.

## Exceptions

### `ArgumentNullException`

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

### `InvalidOperationException`

The connection is not in `Opened`, `Connected` or `Disconnected` state and is therefore not ready to use or the connection is in `Faulted` state and cannot longer be used.

### `ObjectDisposedException`

The connection has been disposed of.

# WriteValues<T>(PlcIdentity, IEnumerable<T>)

Writes an array of the **T** which is associated with the **identity** specified.

## C#

```
public void WriteValues<T>(PlcIdentity identity, IEnumerable<T> values)
```

## Parameters

**identity** `PlcIdentity`

The `PlcIdentity` that can be either a `PlcAddress` which addresses the data area to write. Or a `PlcName` representing the symbolic name of the `PlcType` (to write) associated with the `Device` of this connection.

**values** `IEnumerable<T>`

The array to write to the data area identified by the **identity** specified.

## Exceptions

### `ArgumentException`

The metadata specified could not be resolved to a valid `PlcArrayType` or the **identity** does not refer to a `PlcType` that can be used to access the framework type specified by **T**.

### `ArgumentNullException`

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

### `ArgumentOutOfRangeException`

The number of items in **values** is outside of the bounds defined by `MinLength` and `MaxLength`.

### `InvalidOperationException`



The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

#### ObjectDisposedException

The connection has been disposed of.

## WriteValues<T>(PlcIdentity, T[])

Writes an array of the **T** which is associated with the **identity** specified.

#### C#

```
public void WriteValues<T>(PlcIdentity identity, params T[] values)
```

#### Parameters

**identity** [PlcIdentity](#)

The [PlcIdentity](#) that can be either a [PlcAddress](#) which addresses the data area to write. Or a [PlcName](#) representing the symbolic name of the [PlcType](#) (to write) associated with the [Device](#) of this connection.

**values** **T**

The array to write to the data area identified by the **identity** specified.

#### Exceptions

##### ArgumentException

The metadata specified could not be resolved to a valid [PlcArrayType](#) or the **identity** does not refer to a [PlcType](#) that can be used to access the framework type specified by **T**.

##### ArgumentNullException

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

##### ArgumentOutOfRangeException

The number of items in **values** is outside of the bounds defined by [MinLength](#) and [MaxLength](#).

##### InvalidOperationException

The connection is not in [Opened](#), [Connected](#) or [Disconnected](#) state and is therefore not ready to use or the connection is in [Faulted](#) state and cannot longer be used.

##### ObjectDisposedException

The connection has been disposed of.

## WriteValues<T>(String, IEnumerable<T>)

Writes an array of the **T** which is associated with the **identity** specified.

#### C#

```
public void WriteValues<T>(string identity, IEnumerable<T> values)
```

## Parameters

**identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** IEnumerable<T>

The array to write to the data area identified by the **identity** specified.

## Exceptions

**ArgumentException**

The **identity** is an empty string or the metadata specified could not be resolved to a valid **PlcArrayType** or the **identity** does not refer to a **PlcType** that can be used to access the framework type specified by **T**.

**ArgumentNullException**

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

**ArgumentOutOfRangeException**

The number of items in **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

**InvalidOperationException**

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

**ObjectDisposedException**

The connection has been disposed of.

# WriteValues<T>(String, T[])

Writes an array of the **T** which is associated with the **identity** specified.

**C#**

```
public void WriteValues<T>(string identity, params T[] values)
```

## Parameters

**identity** String

The **PlcIdentity** that can be either a **PlcAddress** which addresses the data area to write. Or a **PlcName** representing the symbolic name of the **PlcType** (to write) associated with the **Device** of this connection.

**values** T

The array to write to the data area identified by the **identity** specified.

## Exceptions

### ArgumentException

The **identity** is an empty string or the metadata specified could not be resolved to a valid **PlcArrayType** or the **identity** does not refer to a **PlcType** that can be used to access the framework type specified by **T**.

### ArgumentNullException

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

### ArgumentOutOfRangeException

The number of items in **values** is outside of the bounds defined by **MinLength** and **MaxLength**.

### InvalidOperationException

The connection is not in **Opened**, **Connected** or **Disconnected** state and is therefore not ready to use or the connection is in **Faulted** state and cannot longer be used.

### ObjectDisposedException

The connection has been disposed of.

## WriteValuesCore(IEnumerable<IPlcValue>)

Writes the data areas described by the **values** specified.

### C#

```
protected virtual void WriteValuesCore(IEnumerable<IPlcValue> values)
```

### Parameters

**values** **IEnumerable<IPlcValue>**

The **IPlcValue**'s to write.

## WriteValuesCore(IPlcValue[])

Writes the data areas described by the **values** specified.

### C#

```
protected virtual void WriteValuesCore(params IPlcValue[] values)
```

### Parameters

**values** **IPlcValue[]**

The **IPlcValue**'s to write.

# WriteValuesCore<T>(PlcArrayType, IEnumerable<T>)

Writes the data area described by the **type** specified.

**C#**

```
protected virtual void WriteValuesCore<T>(PlcArrayType type, IEnumerable<T> values)
```

## Parameters

**type** PlcArrayType

The PlcArrayType to write.

**values** IEnumerable<T>

The values data to write.

# WriteValuesCore<T>(PlcArrayType, T[])

Writes the data area described by the **type** specified.

**C#**

```
protected virtual void WriteValuesCore<T>(PlcArrayType type, params T[] values)
```

## Parameters

**type** PlcArrayType

The PlcArrayType to write.

**values** T

The values data to write.

# Table of Contents

|                                 |    |
|---------------------------------|----|
| <b>Constructors</b>             | 1  |
| PlcDeviceConnection(IPlcDevice) | 1  |
| <b>Fields</b>                   | 1  |
| DefaultTimeout                  | 1  |
| <b>Events</b>                   | 1  |
| Closed                          | 2  |
| Closing                         | 2  |
| Connected                       | 2  |
| Connecting                      | 2  |
| Disconnected                    | 2  |
| Faulted                         | 2  |
| Opened                          | 3  |
| Opening                         | 3  |
| StateChanged                    | 3  |
| <b>Properties</b>               | 3  |
| BreakDetectionTimeout           | 3  |
| Channel                         | 4  |
| ConnectTimeout                  | 4  |
| Device                          | 5  |
| IsConnected                     | 5  |
| ReceiveTimeout                  | 5  |
| State                           | 6  |
| Status                          | 6  |
| TransmitTimeout                 | 6  |
| UseBreakDetection               | 7  |
| UsePool                         | 7  |
| <b>Methods</b>                  | 7  |
| ClearAllPools()                 | 7  |
| ClearPool(PlcDeviceConnection)  | 8  |
| Close()                         | 8  |
| CloseCore()                     | 9  |
| Connect()                       | 9  |
| ConnectCore()                   | 9  |
| CreateChannel()                 | 9  |
| CreateNode(PlcObject)           | 10 |
| CreateNode(PlcType)             | 10 |
| CreateNode(PlcType, IPlcValue)  | 11 |
| DenyIfFaulted()                 | 11 |
| DenyIfIsDisposed()              | 12 |
| DenyIfNotReady()                | 12 |
| Dispose()                       | 12 |
| Dispose(Boolean)                | 12 |
| Finalize()                      | 12 |
| GetStatus(PlcType)              | 13 |
| OnClosed(EventArgs)             | 13 |
| OnClosing(EventArgs)            | 13 |
| OnConnected(EventArgs)          | 14 |
| OnConnecting(EventArgs)         | 14 |
| OnDisconnected(EventArgs)       | 14 |
| OnFaulted(EventArgs)            | 15 |
| OnOpened(EventArgs)             | 15 |

|                                                          |    |
|----------------------------------------------------------|----|
| OnOpening(EventArgs)                                     | 15 |
| OnStateChanged(PlcDeviceConnectionStateChangedEventArgs) | 15 |
| Open()                                                   | 16 |
| OpenCore()                                               | 16 |
| Read(IEnumerable<PlcIdentity>)                           | 16 |
| Read(IEnumerable<PlcType>)                               | 17 |
| Read(IEnumerable<String>)                                | 18 |
| Read(PlcIdentity[])                                      | 18 |
| Read(PlcType[])                                          | 19 |
| Read(String[])                                           | 20 |
| ReadBoolean(PlcBooleanArrayType)                         | 21 |
| ReadBoolean(PlcBooleanType)                              | 21 |
| ReadBoolean(PlcIdentity)                                 | 22 |
| ReadBoolean(PlcIdentity, Int32)                          | 23 |
| ReadBoolean(String)                                      | 24 |
| ReadBoolean(String, Int32)                               | 24 |
| ReadBooleanCore(PlcType)                                 | 25 |
| ReadByte(PlcByteArrayType)                               | 26 |
| ReadByte(PlcByteType)                                    | 26 |
| ReadByte(PlcIdentity)                                    | 27 |
| ReadByte(PlcIdentity, Int32)                             | 28 |
| ReadByte(String)                                         | 29 |
| ReadByte(String, Int32)                                  | 29 |
| ReadByteCore(PlcType)                                    | 30 |
| ReadChar(PlcCharArrayType)                               | 31 |
| ReadChar(PlcCharType)                                    | 31 |
| ReadChar(PlcIdentity)                                    | 32 |
| ReadChar(PlcIdentity, Int32)                             | 33 |
| ReadChar(String)                                         | 34 |
| ReadChar(String, Int32)                                  | 34 |
| ReadCharCore(PlcType)                                    | 35 |
| ReadCore(IEnumerable<PlcType>)                           | 36 |
| ReadCore(PlcType[])                                      | 36 |
| ReadDate(PlcDateType)                                    | 36 |
| ReadDate(PlcIdentity)                                    | 37 |
| ReadDate(String)                                         | 38 |
| ReadDateCore(PlcType)                                    | 39 |
| ReadDateTime(PlcDateTimeType)                            | 39 |
| ReadDateTime(PlcIdentity)                                | 40 |
| ReadDateTime(String)                                     | 40 |
| ReadDateTimeCore(PlcType)                                | 41 |
| ReadDouble(PlcDoubleArrayType)                           | 42 |
| ReadDouble(PlcDoubleType)                                | 42 |
| ReadDouble(PlcIdentity)                                  | 43 |
| ReadDouble(PlcIdentity, Int32)                           | 44 |
| ReadDouble(String)                                       | 45 |
| ReadDouble(String, Int32)                                | 45 |
| ReadDoubleCore(PlcType)                                  | 46 |
| ReadEntities(IEnumerable<IPlcEntity>)                    | 47 |
| ReadEntities(IPlcEntity[])                               | 47 |
| ReadInt16(PlcIdentity)                                   | 48 |
| ReadInt16(PlcIdentity, Int32)                            | 49 |
| ReadInt16(PlcInt16ArrayType)                             | 50 |

|                                               |    |
|-----------------------------------------------|----|
| ReadInt16(PlcInt16Type) .....                 | 50 |
| ReadInt16(String) .....                       | 51 |
| ReadInt16(String, Int32) .....                | 52 |
| ReadInt16Core(PlcType) .....                  | 53 |
| ReadInt32(PlcIdentity) .....                  | 53 |
| ReadInt32(PlcIdentity, Int32) .....           | 54 |
| ReadInt32(PlcInt32ArrayType) .....            | 55 |
| ReadInt32(PlcInt32Type) .....                 | 55 |
| ReadInt32(String) .....                       | 56 |
| ReadInt32(String, Int32) .....                | 57 |
| ReadInt32Core(PlcType) .....                  | 58 |
| ReadInt64(PlcIdentity) .....                  | 58 |
| ReadInt64(PlcIdentity, Int32) .....           | 59 |
| ReadInt64(PlcInt64ArrayType) .....            | 60 |
| ReadInt64(PlcInt64Type) .....                 | 60 |
| ReadInt64(String) .....                       | 61 |
| ReadInt64(String, Int32) .....                | 62 |
| ReadInt64Core(PlcType) .....                  | 63 |
| ReadLReal(PlcIdentity) .....                  | 63 |
| ReadLReal(PlcIdentity, Int32) .....           | 64 |
| ReadLReal(PlcLRealArrayType) .....            | 65 |
| ReadLReal(PlcLRealType) .....                 | 65 |
| ReadLReal(String) .....                       | 66 |
| ReadLReal(String, Int32) .....                | 67 |
| ReadLRealCore(PlcType) .....                  | 68 |
| ReadNodeCore(PlcDataNode) .....               | 68 |
| ReadNodesCore(IEnumerable<PlcDataNode>) ..... | 68 |
| ReadNodesCore(PlcDataNode[]) .....            | 69 |
| ReadObject(PlcIdentity) .....                 | 69 |
| ReadObject(PlcObjectType) .....               | 70 |
| ReadObject(String) .....                      | 71 |
| ReadObject<T>() .....                         | 71 |
| ReadObject<T>(T) .....                        | 72 |
| ReadReal(PlcIdentity) .....                   | 72 |
| ReadReal(PlcIdentity, Int32) .....            | 73 |
| ReadReal(PlcRealArrayType) .....              | 74 |
| ReadReal(PlcRealType) .....                   | 75 |
| ReadReal(String) .....                        | 75 |
| ReadReal(String, Int32) .....                 | 76 |
| ReadRealCore(PlcType) .....                   | 77 |
| ReadS5Time(PlcIdentity) .....                 | 77 |
| ReadS5Time(PlcS5TimeType) .....               | 78 |
| ReadS5Time(String) .....                      | 79 |
| ReadS5TimeCore(PlcType) .....                 | 80 |
| ReadString(PlcIdentity) .....                 | 80 |
| ReadString(PlcIdentity, Int32) .....          | 81 |
| ReadString(PlcStringType) .....               | 82 |
| ReadString(String) .....                      | 82 |
| ReadString(String, Int32) .....               | 83 |
| ReadStringCore(PlcType) .....                 | 84 |
| ReadTime(PlcIdentity) .....                   | 84 |
| ReadTime(PlcTimeType) .....                   | 85 |
| ReadTime(String) .....                        | 86 |

|                                                       |     |
|-------------------------------------------------------|-----|
| ReadTimeCore(PlcType)                                 | 86  |
| ReadTimeOfDay(PlcIdentity)                            | 87  |
| ReadTimeOfDay(PlcTimeOfDayType)                       | 88  |
| ReadTimeOfDay(String)                                 | 88  |
| ReadTimeOfDayCore(PlcType)                            | 89  |
| ReadUInt16(PlcIdentity)                               | 89  |
| ReadUInt16(PlcIdentity, Int32)                        | 90  |
| ReadUInt16(PlcUInt16ArrayType)                        | 91  |
| ReadUInt16(PlcUInt16Type)                             | 92  |
| ReadUInt16(String)                                    | 92  |
| ReadUInt16(String, Int32)                             | 93  |
| ReadUInt16Core(PlcType)                               | 94  |
| ReadUInt32(PlcIdentity)                               | 94  |
| ReadUInt32(PlcIdentity, Int32)                        | 95  |
| ReadUInt32(PlcUInt32ArrayType)                        | 96  |
| ReadUInt32(PlcUInt32Type)                             | 97  |
| ReadUInt32(String)                                    | 97  |
| ReadUInt32(String, Int32)                             | 98  |
| ReadUInt32Core(PlcType)                               | 99  |
| ReadUInt64(PlcIdentity)                               | 99  |
| ReadUInt64(PlcIdentity, Int32)                        | 100 |
| ReadUInt64(PlcUInt64ArrayType)                        | 101 |
| ReadUInt64(PlcUInt64Type)                             | 102 |
| ReadUInt64(String)                                    | 102 |
| ReadUInt64(String, Int32)                             | 103 |
| ReadUInt64Core(PlcType)                               | 104 |
| ReadValue<T>(IPlcValue<T>)                            | 104 |
| ReadValue<T>(PlcIdentity)                             | 105 |
| ReadValue<T>(PlcIdentity, Int32)                      | 106 |
| ReadValue<T>(String)                                  | 107 |
| ReadValue<T>(String, Int32)                           | 107 |
| ReadValueCore<T>(PlcArrayType)                        | 108 |
| ReadValueCore<T>(PlcValueType)                        | 109 |
| ReadValues(IEnumerable<IPlcValue>)                    | 109 |
| ReadValues(IEnumerable<PlcIdentity>)                  | 110 |
| ReadValues(IEnumerable<PlcType>)                      | 110 |
| ReadValues(IEnumerable<String>)                       | 111 |
| ReadValues(IPlcValue[])                               | 112 |
| ReadValues(PlcIdentity[])                             | 113 |
| ReadValues(PlcType[])                                 | 113 |
| ReadValues(String[])                                  | 114 |
| ReadValues<T>(IEnumerable<IPlcValue<T>>)              | 115 |
| ReadValues<T>(IPlcValue<T>[])                         | 115 |
| ReadValues<T>(PlcIdentity, Int32)                     | 116 |
| ReadValues<T>(String, Int32)                          | 117 |
| ReadValuesCore(IEnumerable<IPlcValue>)                | 118 |
| ReadValuesCore(IPlcValue[])                           | 118 |
| ReadValuesCore<T>(PlcArrayType)                       | 119 |
| Write(IEnumerable<KeyValuePair<PlcIdentity, Object>>) | 119 |
| Write(IEnumerable<KeyValuePair<PlcType, Object>>)     | 120 |
| Write(IEnumerable<KeyValuePair<String, Object>>)      | 120 |
| Write(KeyValuePair<PlcIdentity, Object>[])            | 121 |
| Write(KeyValuePair<PlcType, Object>[])                | 122 |



|                                                               |     |
|---------------------------------------------------------------|-----|
| Write(KeyValuePair<String, Object>[]) .....                   | 122 |
| WriteBoolean(PlcBooleanArrayType, Boolean[]) .....            | 123 |
| WriteBoolean(PlcBooleanArrayType, IEnumerable<Boolean>) ..... | 124 |
| WriteBoolean(PlcBooleanType, Boolean) .....                   | 124 |
| WriteBoolean(PlcIdentity, Boolean) .....                      | 125 |
| WriteBoolean(PlcIdentity, Boolean[]) .....                    | 126 |
| WriteBoolean(PlcIdentity, IEnumerable<Boolean>) .....         | 126 |
| WriteBoolean(String, Boolean) .....                           | 127 |
| WriteBoolean(String, Boolean[]) .....                         | 128 |
| WriteBoolean(String, IEnumerable<Boolean>) .....              | 129 |
| WriteBooleanCore(PlcType, Boolean[]) .....                    | 129 |
| WriteByte(PlcByteArrayType, Byte[]) .....                     | 130 |
| WriteByte(PlcByteArrayType, IEnumerable<Byte>) .....          | 130 |
| WriteByte(PlcByteType, Byte) .....                            | 131 |
| WriteByte(PlcIdentity, Byte) .....                            | 132 |
| WriteByte(PlcIdentity, Byte[]) .....                          | 132 |
| WriteByte(PlcIdentity, IEnumerable<Byte>) .....               | 133 |
| WriteByte(String, Byte) .....                                 | 134 |
| WriteByte(String, Byte[]) .....                               | 134 |
| WriteByte(String, IEnumerable<Byte>) .....                    | 135 |
| WriteByteCore(PlcType, Byte[]) .....                          | 136 |
| WriteChar(PlcCharArrayType, Char[]) .....                     | 136 |
| WriteChar(PlcCharArrayType, IEnumerable<Char>) .....          | 137 |
| WriteChar(PlcCharType, Char) .....                            | 138 |
| WriteChar(PlcIdentity, Char) .....                            | 138 |
| WriteChar(PlcIdentity, Char[]) .....                          | 139 |
| WriteChar(PlcIdentity, IEnumerable<Char>) .....               | 140 |
| WriteChar(String, Char) .....                                 | 140 |
| WriteChar(String, Char[]) .....                               | 141 |
| WriteChar(String, IEnumerable<Char>) .....                    | 142 |
| WriteCharCore(PlcType, Char[]) .....                          | 143 |
| WriteCore(IEnumerable<KeyValuePair<PlcType, Object>>) .....   | 143 |
| WriteDate(PlcDateType, DateTime) .....                        | 143 |
| WriteDate(PlcIdentity, DateTime) .....                        | 144 |
| WriteDate(String, DateTime) .....                             | 145 |
| WriteDateCore(PlcType, DateTime) .....                        | 145 |
| WriteDateTime(PlcDateTimeType, DateTime) .....                | 146 |
| WriteDateTime(PlcIdentity, DateTime) .....                    | 146 |
| WriteDateTime(String, DateTime) .....                         | 147 |
| WriteDateTimeCore(PlcType, DateTime) .....                    | 148 |
| WriteDouble(PlcDoubleArrayType, Double[]) .....               | 148 |
| WriteDouble(PlcDoubleArrayType, IEnumerable<Double>) .....    | 149 |
| WriteDouble(PlcDoubleType, Double) .....                      | 149 |
| WriteDouble(PlcIdentity, Double) .....                        | 150 |
| WriteDouble(PlcIdentity, Double[]) .....                      | 151 |
| WriteDouble(PlcIdentity, IEnumerable<Double>) .....           | 151 |
| WriteDouble(String, Double) .....                             | 152 |
| WriteDouble(String, Double[]) .....                           | 153 |
| WriteDouble(String, IEnumerable<Double>) .....                | 154 |
| WriteDoubleCore(PlcType, Double[]) .....                      | 154 |
| WriteEntities(IEnumerable<IPlcEntity>) .....                  | 155 |
| WriteEntities(IPlcEntity[]) .....                             | 156 |
| WriteInt16(PlcIdentity, IEnumerable<Int16>) .....             | 156 |

|                                                    |     |
|----------------------------------------------------|-----|
| WriteInt16(PlcIdentity, Int16)                     | 157 |
| WriteInt16(PlcIdentity, Int16[])                   | 158 |
| WriteInt16(PlcInt16ArrayType, IEnumerable<Int16>)  | 159 |
| WriteInt16(PlcInt16ArrayType, Int16[])             | 159 |
| WriteInt16(PlcInt16Type, Int16)                    | 160 |
| WriteInt16(String, IEnumerable<Int16>)             | 161 |
| WriteInt16(String, Int16)                          | 161 |
| WriteInt16(String, Int16[])                        | 162 |
| WriteInt16Core(PlcType, Int16[])                   | 163 |
| WriteInt32(PlcIdentity, IEnumerable<Int32>)        | 163 |
| WriteInt32(PlcIdentity, Int32)                     | 164 |
| WriteInt32(PlcIdentity, Int32[])                   | 165 |
| WriteInt32(PlcInt32ArrayType, IEnumerable<Int32>)  | 165 |
| WriteInt32(PlcInt32ArrayType, Int32[])             | 166 |
| WriteInt32(PlcInt32Type, Int32)                    | 167 |
| WriteInt32(String, IEnumerable<Int32>)             | 167 |
| WriteInt32(String, Int32)                          | 168 |
| WriteInt32(String, Int32[])                        | 169 |
| WriteInt32Core(PlcType, Int32[])                   | 170 |
| WriteInt64(PlcIdentity, IEnumerable<Int64>)        | 170 |
| WriteInt64(PlcIdentity, Int64)                     | 171 |
| WriteInt64(PlcIdentity, Int64[])                   | 171 |
| WriteInt64(PlcInt64ArrayType, IEnumerable<Int64>)  | 172 |
| WriteInt64(PlcInt64ArrayType, Int64[])             | 173 |
| WriteInt64(PlcInt64Type, Int64)                    | 174 |
| WriteInt64(String, IEnumerable<Int64>)             | 174 |
| WriteInt64(String, Int64)                          | 175 |
| WriteInt64(String, Int64[])                        | 176 |
| WriteInt64Core(PlcType, Int64[])                   | 176 |
| WriteLReal(PlcIdentity, Double)                    | 177 |
| WriteLReal(PlcIdentity, Double[])                  | 177 |
| WriteLReal(PlcIdentity, IEnumerable<Double>)       | 178 |
| WriteLReal(PlcLRealArrayType, Double[])            | 179 |
| WriteLReal(PlcLRealArrayType, IEnumerable<Double>) | 180 |
| WriteLReal(PlcLRealType, Double)                   | 180 |
| WriteLReal(String, Double)                         | 181 |
| WriteLReal(String, Double[])                       | 182 |
| WriteLReal(String, IEnumerable<Double>)            | 182 |
| WriteLRealCore(PlcType, Double[])                  | 183 |
| WriteNodeCore(PlcDataNode)                         | 184 |
| WriteNodesCore(IEnumerable<PlcDataNode>)           | 184 |
| WriteNodesCore(PlcDataNode[])                      | 184 |
| WriteObject<T>(T)                                  | 184 |
| WriteReal(PlcIdentity, IEnumerable<Single>)        | 185 |
| WriteReal(PlcIdentity, Single)                     | 186 |
| WriteReal(PlcIdentity, Single[])                   | 186 |
| WriteReal(PlcRealArrayType, IEnumerable<Single>)   | 187 |
| WriteReal(PlcRealArrayType, Single[])              | 188 |
| WriteReal(PlcRealType, Single)                     | 189 |
| WriteReal(String, IEnumerable<Single>)             | 189 |
| WriteReal(String, Single)                          | 190 |
| WriteReal(String, Single[])                        | 191 |
| WriteRealCore(PlcType, Single[])                   | 191 |

|                                                            |     |
|------------------------------------------------------------|-----|
| WriteS5Time(PlcIdentity, TimeSpan) .....                   | 192 |
| WriteS5Time(PlcS5TimeType, TimeSpan) .....                 | 192 |
| WriteS5Time(String, TimeSpan) .....                        | 193 |
| WriteS5TimeCore(PlcType, TimeSpan) .....                   | 194 |
| WriteString(PlcIdentity, String) .....                     | 194 |
| WriteString(PlcStringType, String) .....                   | 195 |
| WriteString(String, String) .....                          | 195 |
| WriteStringCore(PlcType, String) .....                     | 196 |
| WriteTime(PlcIdentity, TimeSpan) .....                     | 196 |
| WriteTime(PlcTimeType, TimeSpan) .....                     | 197 |
| WriteTime(String, TimeSpan) .....                          | 198 |
| WriteTimeCore(PlcType, TimeSpan) .....                     | 198 |
| WriteTimeOfDay(PlcIdentity, TimeSpan) .....                | 199 |
| WriteTimeOfDay(PlcTimeOfDayType, TimeSpan) .....           | 199 |
| WriteTimeOfDay(String, TimeSpan) .....                     | 200 |
| WriteTimeOfDayCore(PlcType, TimeSpan) .....                | 201 |
| WriteUInt16(PlcIdentity, IEnumerable<UInt16>) .....        | 201 |
| WriteUInt16(PlcIdentity, UInt16) .....                     | 202 |
| WriteUInt16(PlcIdentity, UInt16[]) .....                   | 203 |
| WriteUInt16(PlcUInt16ArrayType, IEnumerable<UInt16>) ..... | 203 |
| WriteUInt16(PlcUInt16ArrayType, UInt16[]) .....            | 204 |
| WriteUInt16(PlcUInt16Type, UInt16) .....                   | 205 |
| WriteUInt16(String, IEnumerable<UInt16>) .....             | 205 |
| WriteUInt16(String, UInt16) .....                          | 206 |
| WriteUInt16(String, UInt16[]) .....                        | 207 |
| WriteUInt16Core(PlcType, UInt16[]) .....                   | 208 |
| WriteUInt32(PlcIdentity, IEnumerable<UInt32>) .....        | 208 |
| WriteUInt32(PlcIdentity, UInt32) .....                     | 209 |
| WriteUInt32(PlcIdentity, UInt32[]) .....                   | 209 |
| WriteUInt32(PlcUInt32ArrayType, IEnumerable<UInt32>) ..... | 210 |
| WriteUInt32(PlcUInt32ArrayType, UInt32[]) .....            | 211 |
| WriteUInt32(PlcUInt32Type, UInt32) .....                   | 212 |
| WriteUInt32(String, IEnumerable<UInt32>) .....             | 212 |
| WriteUInt32(String, UInt32) .....                          | 213 |
| WriteUInt32(String, UInt32[]) .....                        | 214 |
| WriteUInt32Core(PlcType, UInt32[]) .....                   | 214 |
| WriteUInt64(PlcIdentity, IEnumerable<UInt64>) .....        | 215 |
| WriteUInt64(PlcIdentity, UInt64) .....                     | 216 |
| WriteUInt64(PlcIdentity, UInt64[]) .....                   | 216 |
| WriteUInt64(PlcUInt64ArrayType, IEnumerable<UInt64>) ..... | 217 |
| WriteUInt64(PlcUInt64ArrayType, UInt64[]) .....            | 218 |
| WriteUInt64(PlcUInt64Type, UInt64) .....                   | 218 |
| WriteUInt64(String, IEnumerable<UInt64>) .....             | 219 |
| WriteUInt64(String, UInt64) .....                          | 220 |
| WriteUInt64(String, UInt64[]) .....                        | 220 |
| WriteUInt64Core(PlcType, UInt64[]) .....                   | 221 |
| WriteValue<T>(IPlcValue<T>) .....                          | 222 |
| WriteValue<T>(PlcArrayType, IEnumerable<T>) .....          | 222 |
| WriteValue<T>(PlcArrayType, T[]) .....                     | 223 |
| WriteValue<T>(PlcIdentity, IEnumerable<T>) .....           | 224 |
| WriteValue<T>(PlcIdentity, T) .....                        | 224 |
| WriteValue<T>(PlcIdentity, T[]) .....                      | 225 |
| WriteValue<T>(PlcValueType, T) .....                       | 226 |

|                                                        |     |
|--------------------------------------------------------|-----|
| WriteValue<T>(String, IEnumerable<T>) .....            | 226 |
| WriteValue<T>(String, T) .....                         | 227 |
| WriteValue<T>(String, T[]) .....                       | 228 |
| WriteValueCore<T>(PlcArrayType, IEnumerable<T>) .....  | 229 |
| WriteValueCore<T>(PlcArrayType, T[]) .....             | 229 |
| WriteValueCore<T>(PlcValueType, T) .....               | 229 |
| WriteValues(IEnumerable<IPlcValue>) .....              | 230 |
| WriteValues(IPlcValue[]) .....                         | 230 |
| WriteValues<T>(IEnumerable<IPlcValue<T>>) .....        | 231 |
| WriteValues<T>(IPlcValue<T>[]) .....                   | 231 |
| WriteValues<T>(PlcIdentity, IEnumerable<T>) .....      | 232 |
| WriteValues<T>(PlcIdentity, T[]) .....                 | 233 |
| WriteValues<T>(String, IEnumerable<T>) .....           | 233 |
| WriteValues<T>(String, T[]) .....                      | 234 |
| WriteValuesCore(IEnumerable<IPlcValue>) .....          | 235 |
| WriteValuesCore(IPlcValue[]) .....                     | 235 |
| WriteValuesCore<T>(PlcArrayType, IEnumerable<T>) ..... | 236 |
| WriteValuesCore<T>(PlcArrayType, T[]) .....            | 236 |