PlcDeviceConnectionChannel Members

Namespace: IPS7Lnk.Advanced
Assemblies: IPS7LnkNet.Advanced.dll, IPS7LnkNet.Advanced.dll
The PlcDeviceConnectionChannel type exposes the following members.

Constructors

PlcDeviceConnectionChannel(PlcDeviceConnection)

Initializes a new instance of the PlcDeviceConnectionChannel class using the specified `connection`.

C#

```
protected PlcDeviceConnectionChannel(PlcDeviceConnection connection)
```

Parameters

`connection` PlcDeviceConnection

The PlcDeviceConnection from that the channel configuration is to be adopted.

Exceptions

ArgumentNullException

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

Properties

Address

Gets the IP address of the endpoint used.

C#

```
public string Address { get; }
```

Property Value

String

A String representing the IP address of the endpoint used.

BreakDetectionTimeout

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

C#

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

Property Value

Int32

The time in milliseconds to detect a connection break.

ConnectTimeout

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

C#

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

Property Value

Int32

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

IsConnected

When implemented in a derived class, gets a value indicating whether the connection has been completly established to the device.

C#

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

Property Value

Boolean

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

LocalTSAP

Gets the address of the local transport service access point used when communicating with the PLC device.

C#

```
public byte[] LocalTSAP { get; }
```

Property Value

Byte[]

An array of Byte values identifying the local TSAP used.

Rack

Gets the rack number of the endpoint used.

C#

```
public int Rack { get; }
```

Property Value

Int32

An integer value in the range MinRack to MaxRack indicating the rack number of the endpoint used.

ReceiveTimeout

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

C#

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

Property Value

Int32

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

RemoteTSAP

Gets or sets the address of the remote transport service access point used when communicating with the PLC device.

C#

```
public byte[] RemoteTSAP { get; }
```

Property Value

Byte[]

An array of Byte values identifying the remote TSAP used.

Slot

Gets the slot number of the endpoint used.

C#

```
public int Slot { get; }
```

Property Value

Int32

An integer value in the range MinSlot to MaxSlot indicating the slot number of the endpoint used.

SyncRoot

Gets an object that can be used to synchronize access to the PlcDeviceConnectionChannel.

C#

```
public object SyncRoot { get; }
```

Property Value

Object

An object that can be used to synchronize access to the PlcDeviceConnectionChannel.

TransmitTimeout

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

C#

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

Property Value

Int32

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

UseBreakDetection

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

C#

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

Property Value

Boolean

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

Methods

Close()

Closes an established connection to a device.

C#

```
public PlcStatus Close()
```

Returns

PlcStatus

A PlcStatus instance which describes the outcome of the operation.

CloseCore()

When implemented in a derived class, closes an established connection to a device.

C#

```
protected abstract PlcStatus CloseCore()
```

Returns

PlcStatus

A PlcStatus instance which describes the outcome of the operation.

Connect()

Fully establishes a connection to a device.

C#

```
public PlcStatus Connect()
```

Returns

PlcStatus

A PlcStatus instance which describes the outcome of the operation.

Exceptions

ObjectDisposedException

The channel has been disposed of.

ConnectCore()

When implemented in a derived class, fully establishes a connection to a device.

C#

```
protected abstract PlcStatus ConnectCore()
```

Returns

PlcStatus

A PlcStatus instance which describes the outcome of the operation.

DenyIfIsDisposed()

Verifies whether the channel has been disposed of.

C#

```
protected void DenyIfIsDisposed()
```

Exceptions

ObjectDisposedException

The channel has been disposed of.

Dispose()

Releases all resources used by the PlcDeviceConnectionChannel.

C#

```
public void Dispose()
```

Dispose(Boolean)

Releases the unmanaged resources used by the PlcDeviceConnectionChannel 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 PlcDeviceConnectionChannel class.

C#

```
protected void Finalize()
```

HasConfigurationOf(PlcDeviceConnection)

Evaluates the setup of the `connection` specified to determine whether the configuration of this PlcDeviceConnectionChannel matches with the settings of the channel.

C#

```
public bool HasConfigurationOf(PlcDeviceConnection connection)
```

Parameters

`connection` PlcDeviceConnection

The PlcDeviceConnection its setup is to compared with the setup of the channel.

Returns

Boolean

The value true if the setup of the channel matches the setup of the `connection` specified; otherwise the value false.

Remarks

Using this method the framework determines if an existing PlcDeviceConnectionChannel can be used for the `connection` specified. By default this method evaluates a used IPDeviceEndPoint instance, its Address, Rack, Slot, LocalTSAP and RemoteTSAP.

HasConfigurationOfCore(PlcDeviceConnection)

Performs additional custom evaluation of the setup of the `connection` specified to determine whether the configuration of a derivat of the PlcDeviceConnectionChannel matches with the settings of the channel.

C#

```
protected virtual bool HasConfigurationOfCore(PlcDeviceConnection connection)
```

Parameters

`connection` PlcDeviceConnection

The PlcDeviceConnection its setup is to compared with the setup of the channel.

Returns

Boolean

The value true if the setup of the channel matches the setup of the `connection` specified; otherwise the value false. The default value returned by this method is true.

Open()

Establishes a connection to a device.

C#

```
public PlcStatus Open()
```

Returns

PlcStatus

A PlcStatus instance which describes the outcome of the operation.

Exceptions

ObjectDisposedException

The channel has been disposed of.

OpenCore()

When implemented in a derived class, establishes a connection to a device.

C#

```
protected abstract PlcStatus OpenCore()
```

Returns

PlcStatus

A PlcStatus instance which describes the outcome of the operation.

Table of Contents