

# **Beckhoff TwinCAT Driver Help**

© 2009 Kepware Technologies

# Table of Contents

<b>1</b>	<b>Getting Started</b> .....	<b>3</b>
	Help Contents.....	3
	Overview .....	3
<b>2</b>	<b>Device Setup</b> .....	<b>4</b>
	Device Setup .....	4
	Symbolic Settings.....	5
	Setting Up an AMS Remote Connection.....	7
<b>3</b>	<b>Automatic Tag Database Generation</b> .....	<b>11</b>
	Automatic Tag Database Generation.....	11
<b>4</b>	<b>Optimizing Beckhoff TwinCat Communications</b> .....	<b>12</b>
	Performance Optimization.....	12
<b>5</b>	<b>Data Types Description</b> .....	<b>13</b>
	Data Types Description.....	13
<b>6</b>	<b>Address Descriptions</b> .....	<b>14</b>
	Address Descriptions.....	14
	Symbolic Tag Based Addressing.....	14
	Tag Scope .....	14
	Addressing Atomic Data Types.....	15
	Ordering of TwinCat Array Data.....	16
<b>7</b>	<b>Error Descriptions</b> .....	<b>17</b>
	Error Descriptions.....	17
	Error Codes .....	18
	Address Validation.....	20
	Address Validation.....	20
	Device Address <address> contains a syntax error.....	20
	Data type '<type>' is not valid for device address '<address>'.....	20
	Array size is out of range for address '<address>'.....	21
	Address '<address>' is out of range for the specified device or register.....	21
	Driver Error Messages.....	21
	Driver Error Messages.....	21
	Unable to load '<dll>'.....	21
	Unable to import from '<dll>'.....	21
	Port open failed with Error Code '<#>'.....	22
	Driver Status Error Messages.....	22
	Device Status Error Messages.....	22
	Device '<Device name>' is not responding.....	22
	Unable to synchronize with configuration file '<file name>' on device '<device name>' due to memory allocation error.....	22
	Unable to perform Beckhoff compatibility on device '<device name>' due to memory allocation error.....	23
	Unable to gather runtime information for device '<device name>'. Reason '<reason>'.....	23
	Read Error Messages.....	23
	Read Error Messages.....	23
	Unable to read tag '<address>' on device '<device name>'. Error Code '<#>'.....	23
	Unable to read tag '<address>' on device '<device name>'. Tag data size of '<size>' bytes(s) exceeds actual data size of '<size>' byte(s).....	24
	Unable to read tag '<address>' on device '<device name>'. Address bounds exceeded.....	24
	Unable to read tag '<address>' on device '<device name>'. Memory allocation error.....	24

Unable to read tag '<address>' on device '<device name>'. Actual data type is not compatible with tag of type '<data type>'.....	24
Unable to read tag '<address>' on device '<device name>'. Runtime file is not valid.....	25
Unable to read tag '<address>' on device '<device name>'. Symbol not found in file.....	25
Unable to read tag '<address>' on device '<device name>'. Bekchoff TwinCAT DLL(s) necessary for runtime are not loaded.....	25
Unable to read tag '<address>' on device '<device name>'. Tag does not meet filtering requirements.....	25
<b>Write Error Messages.....</b>	<b>25</b>
Write Error Messages.....	25
Unable to write to tag '<address>' on device '<device name>'. Error Code '<#>'.....	26
Unable to write to tag '<address>' on device '<device name>'. Tag data size of '<size>' bytes(s) exceeds actual data size of '<size>' byte(s).....	26
Unable to write to tag '<address>' on device '<device name>'. Address bounds exceeded.....	26
Unable to write to tag '<address>' on device '<device name>'. Memory allocation error.....	27
Unable to write to tag '<address>' on device '<device name>'. Actual data type is not compatible with tag of type '<data type>'.....	27
Unable to write to tag '<address>' on device '<device name>'. Runtime file is not valid.....	27
Unable to write to tag '<address>' on device '<device name>'. Symbol not found in file.....	27
Unable to write to tag '<address>' on device '<device name>'. Bekchoff TwinCAT DLL(s) necessary for runtime are not loaded.....	27
Unable to write to tag '<address>' on device '<device name>'. Tag does not meet filtering requirements.....	28
Unable to write to tag '<address>' on device '<device name>'. Tag access is Read Only.....	28
<b>Automatic Tag Database Generation Messages.....</b>	<b>28</b>
Automatic Tag Database Generation Error Messages.....	28
Unable to generate a tag database for device '<device name>'. Reason: Memory allocation error.....	28
Unable to generate a tag database for device '<device name>'. Reason: Beckhoff TwinCAT DLL(s) necessary for import are not loaded.....	29
Unable to generate a tag database for device '<device name>'. Reason: Device returned error code '<#>'.....	29
Unable to generate a tag database for device '<device name>'. Reason: Device is not responding.....	29
<b>8 Technical Notes.....</b>	<b>29</b>
<b>Technical Notes.....</b>	<b>29</b>
<b>TwinCAT Memory Warning.....</b>	<b>30</b>
<b>TwinCAT Time Slice Notes.....</b>	<b>30</b>

**Index**

## Beckhoff TwinCAT Driver Help

---

Help version 1.034

### CONTENTS

#### [Overview](#)

What is the Beckhoff TwinCAT Device Driver?

#### [Device Setup](#)

How do I configure a device for use with this driver?

#### [Automatic Tag Database Generation](#)

How can I easily configure tags for the Beckhoff TwinCAT driver?

#### [Optimizing Your Beckhoff TwinCAT Communications](#)

How do I get the best performance from the Beckhoff TwinCAT driver?

#### [Data Types Description](#)

What data types does the Beckhoff TwinCAT driver support?

#### [Address Descriptions](#)

How do I reference a data location in a Beckhoff TwinCAT device?

#### [Error Descriptions](#)

What error messages does the Beckhoff TwinCAT driver produce?

#### [Technical Notes](#)

Where can I find technical notes for the Beckhoff TwinCAT Device Driver?

## Overview

---

The Beckhoff TwinCAT Driver is specifically designed to communicate with a Beckhoff TwinCAT software system. The Beckhoff TwinCAT software system turns any compatible PC into a real-time controller with a multi-PLC system, NC axis control, programming environment and operating station. TwinCAT replaces conventional PLC and NC/CNC controllers as well as operating devices with:

- open, compatible PC hardware
- embedded IEC 61131-3 software PLC, software NC and software CNC in Windows NT/2000/XP/Vista, NT/ XP Embedded, CE
- programming and run-time systems optionally together on one PC or separated
- connection to all common fieldbuses
- PC interfaces are supported
- data communication with user interfaces and other programs by means of open Microsoft standards (OPC, OCX, DLL, etc.)

### World-wide Connection Through Message Routing

TwinCAT PLC programs can run on PCs or on Beckhoff Bus Terminal Controllers. A **message router** manages and distributes all the messages, both in the system and via TCP/IP connections. PC systems can be connected with each other via TCP/IP; Bus Terminal Controllers are integrated via serial interfaces and fieldbuses.

### TwinCAT ADS (Automation Device Specification)

The data link to TwinCAT servers always takes place via the message router system. This enables Windows programs not only to work with the local server but also to exchange data with all registered TwinCAT servers worldwide. The message router also ensures that the data exchange to remote servers on other PCs or field equipment can occur.

### TwinCAT I/O – Universal I/O Interface For All Common Fieldbuses

Many PC fieldbus cards from various manufacturers are supported. It is possible to operate more than one fieldbus card per PC. Master and slave functionality is supported, depending on the selected fieldbus card. The fieldbus cards can be configured and diagnosed conveniently via the TwinCAT System Manager. TwinCAT I/O includes the TwinCAT real-time system for operating the fieldbuses and a DLL interface to application programs.

## TwinCAT PLC

TwinCAT PLC was conceived as a pure software PLC. It allows up to four **virtual PLC CPUs** on one PC, with each running up to four separate user tasks. TwinCAT PLC includes both the programming environment and the run-time system. Under the CE operating system and the embedded operating systems for the series BX and BC controllers, only TwinCAT run-time is available. Program modifications are implemented via network-capable powerful communication with the run-time system.

## Device Setup

---

### Communication Protocol

ADS API

### Supported Devices

Beckhoff TwinCAT PLC  
BC9xxx Coupler Controller  
BX9xxx Coupler Controller

### Maximum Channels and Devices

The maximum number of channels supported by this driver is 100. The maximum number of devices is 1024 per channel.

### Request Timeout

This parameter specifies the amount of time that the driver will wait for a response from the device before giving up and going on to the next request. Long timeouts will only affect performance if a device is not responding. The default setting is 1000 milliseconds. The valid range is 100 to 30000 milliseconds.

### Retry Attempts

This parameter specifies the number of times that the driver will retry a message before giving up and going on to the next message. The default setting is 3 retries. The valid range is 1 to 10.

### Device IDs

The Device ID is a specific ADS-AMS Net ID. Every PC on the network can be uniquely identified by a TCP/IP address. The ADS-AMS Net ID is an extension of the TCP/IP address and identifies a TwinCAT message router, such as "255.255.255.255.1.1". TwinCAT message routers exist on every TwinCAT PC and on every Beckhoff BCxxxx bus controller.

### AMS Remote Connections Management

**Important:** In order to communicate successfully with any remote TwinCAT runtime engine on a Beckhoff PLC (or an ADS library on any remote PC), the AMS Remote Connection Manager utility **must** be used to assign an AMS Net ID to the local station hosting the Beckhoff driver. The AMS Net ID is also used to establish routes to remote AMS/ADS enabled devices. For more information, refer to [Setting Up an AMS Remote Connection](#).

**Note:** For more information about the AMS messaging protocol, please consult Beckhoff's help documentation.

### Tag Database Creation

The Automatic OPC Tag Database Generation features of this driver have been designed to make setting up the OPC application less time consuming. This driver can be configured to automatically build a list of Server tags within the OPC Server that correspond to device specific data. The automatically generated OPC tags can then be browsed from the OPC client.

### Port Number

The ADS devices in a TwinCAT message router are uniquely identified by a number referred to as the ADS-PortNr. The TwinCAT System Manager supports four different TwinCAT PLC runtime systems at ports 801, 811, 821, and 831. The default port is 801. Other supported ports include port 800 for BC9xxx and BX9xxx series controllers and port 301 and 302 for additional tasks.

**Note:** Users must allow the TCP port number 48898 (AMS port 801) on the firewall if planning to connect remotely to an ADS/AMS router. If not, the system will not work reliably and may stop.

### Default Data Type

This parameter specifies the data type that will be assigned to a Client/Server tag when the default type is selected during tag addition, modification and import. Client/Server tags are assigned the default data type when any of the following conditions occur:

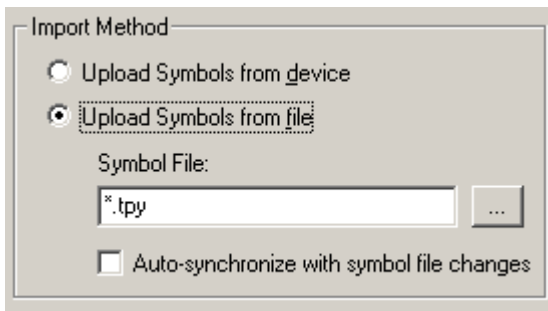
1. A Dynamic Tag is created in the client with Native as its assigned data type.
2. A Static Tag is created in the server with Default as its assigned data type.

## Symbolic Settings

---

### Import Method

The generated server tags are based on the tags defined in the Beckhoff TwinCAT device. There are two database import methods that can be used to create a tag database: Upload Symbols from Device and Upload Symbols from File.



### Upload Symbols from Device

This feature retrieves the tags directly from the controller over the same Ethernet connection used for data access.

**Note:** At this time, tags and symbols can only be uploaded from a TwinCAT soft PLC. They cannot be uploaded from a hardware BC/BX controller.

### Upload Symbols from File

This feature requires that, when creating the tag database from an import file, the import file be a .tpy file generated from the Beckhoff TwinCAT PLC Control software.

### Symbol File

This parameter specifies the exact location of the .tpy file from which tags will be imported. BC9xxx and BX9xxx models must include this file for a runtime database.

### Auto-Synchronize With Symbol File Changes

This feature updates the runtime database automatically when the file has been modified.

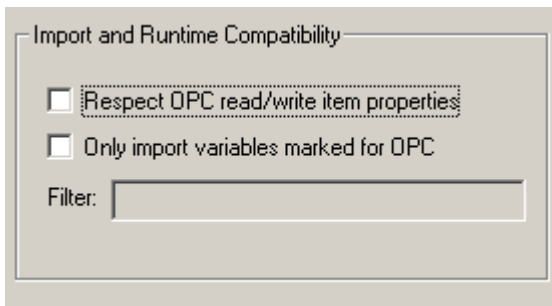
**Caution:** Checking Auto-Synchronize will reflect changes to the runtime, regardless of whether or not the new configuration has been pushed down to the device. Users should ensure that the device is using the same .tpy file as loaded in the server project; otherwise, inaccurate data could be obtained.

**Note 1:** Tag database changes will not appear in the server until Automatic Tag Generation is performed.

**Note 2:** Only the BC9xxx and BX9xxx models support database creation from a .tpy file.

### Import and Runtime Compatibility

Additional OPC item information is located in the comment field after the PLC variable declaration in TwinCAT PLC Control. This optional functionality is used to minimize the OPC namespace of the server project. It also assigns the appropriate Read/Write item properties and OPC Client access.



### Respect OPC Read/Write Item Properties

When enabled, variables that include the string "OPC\_PROP[0005]: <r/w access number>" (defined within the variable comment field in the PLC project) will be assigned the appropriate Read/Write access. During import and runtime, any tag(s) corresponding to a variable with the R/W access specified will carry the same Read/Write access rights. Whitespace is ignored.

- **Syntax:**

(\*~(OPC\_PROP[0005] : <Item Access Rights> : <remark>)\*)

<Item Access Rights> 1 = Read Only, 2 = Write Only, 3 = Read/Write.  
<remark> Optional and ignored.

- **Read Only Example:**

dwTemperature:DWORD; (\*~(OPC\_PROP[0005]:1:Read Only)\*)

### Only Import Variables Marked for OPC

When enabled, only tags with the keyword "OPC:1" (defined within the variable comment field in the PLC project) will be imported and visible during runtime.

- **Syntax:**

(\*~(OPC : 1 : <remark>)\*)

<remark> Optional and ignored.

- **Example:**

bMemFlag AT%MX0.0:BOOL; (\*~(OPC:1:Visible to OPC Clients)\*)

### Filter + Only Import Variables Marked for OPC

When Only Import Variables Marked for OPC is enabled with a Filter string, only tags in which the comment includes the string "OPC:1" and "OPC\_Filter: <filter>" will be imported into the project and visible during runtime. Filter strings are arbitrary, user-defined strings used for another layer of filtering. If none are specified, this field will be ignored. Otherwise, the variable will only be imported if the filter string found in the comment section matches the filter set in Device Properties. Whitespace is treated literally.

- **Syntax:**

(\*~(OPC : 1 : <remark>)(OPC\_Filter : <filter string> : <remark>)\*)

<remark> Optional and ignored.

- **Example:**

bMemFlag AT%MX0.0:BOOL; (\*~(OPC:1:Visible to OPC Clients)(OPC\_Filter:Memory Access)\*)

### Combination Usage

Each of the comments described above can be combined for greater control over the filtering and assignment of OPC Read/Write item properties.

### Only Import Variables Marked for OPC + Respect OPC Read/Write Item Properties

Variables will only be imported and visible during runtime if "OPC:1" is defined in the comment field. Upon being imported, corresponding tags will be assigned the appropriate Read/Write access rights.

- **Syntax:**

(\*~(OPC : 1 : <remark>)(OPC\_PROP[0005] : <Item Access Rights> : <remark>)\*)

- **Example:**

```
dwTemperature:DWORD; (*~(OPC:1:Visible to OPC Clients)(OPC_PROP[0005]:1:Read Only)*)
```

#### Only Import Variables Marked for OPC + Filter + Respect OPC Item Properties

A variable with this comment will only be imported and visible during runtime if "OPC:1" is defined in the comment field and if the filter string found in the comment section matches the filter set in Device Properties. Corresponding tags will be assigned Read/Write access rights once they are imported.

- **Syntax:**

```
(*~(OPC : 1 : <remark>)(OPC_Filter : <filter string> : <remark>) (OPC_PROP[0005] : <Item Access Rights> : <remark>)*)
```

- **Example:**

```
dwTemperature:DWORD; (*~(OPC:1:Visible to OPC Clients) (OPC_Filter:Memory Access) (OPC_PROP[0005]:1:Read Only)*)
```

#### OPC Item Description + Only Import Variables Marked for OPC

Any characters placed before the tilde in the comment field are treated as a description and will be displayed in the OPC Server as an Item Description.

- **Syntax:**

```
(*<OPC Item Description>~(OPC : 1 : <remark>)*)
```

- **Example:**

```
(*This is my OPC Item Description~(OPC :1:Visible to OPC Clients)*)
```

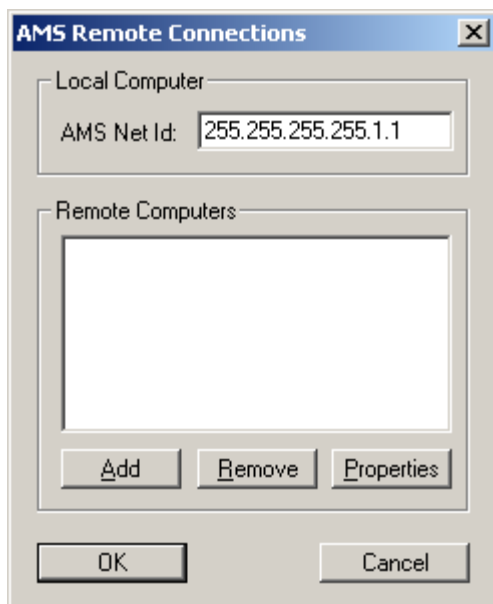
## Setting Up an AMS Remote Connection

For more information, refer to [AMS Remote Connections Management](#).

### Configuration Without TwinCAT PLC Installed

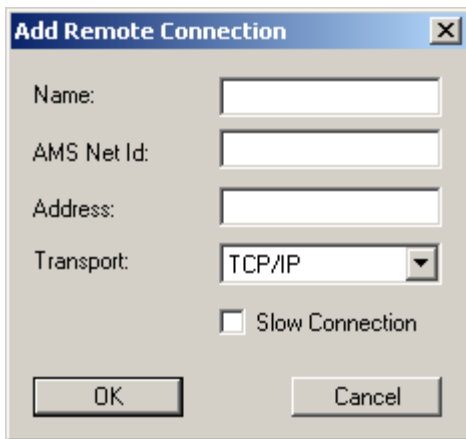
A TwinCAT-PC's ADS-AMS Net ID is set using TwinCAT's Remote Manager Utility.

1. Open the **Start** menu. In the **OPC Server Directory**, locate the **Utilities** folder.
2. Double-click on **TcAmsRemoteMgr.exe**.



3. By default, TwinCAT constructs the **AMS Net ID** from the **TCP/IP** address of the local PC using an extension of ".1.1". The AMS Net ID can, however, be chosen freely.

4. Press **Add** to include a **remote computer** in the connection.

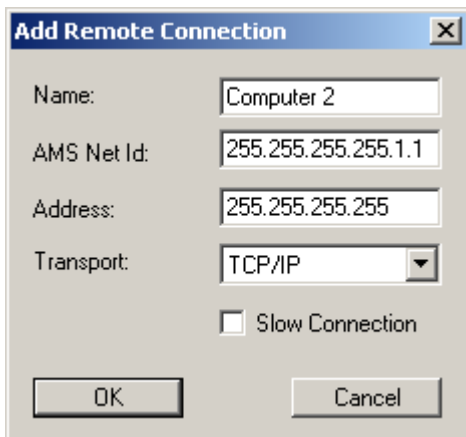


The screenshot shows a dialog box titled "Add Remote Connection" with a close button (X) in the top right corner. It contains the following fields and controls:

- Name: [Empty text box]
- AMS Net Id: [Empty text box]
- Address: [Empty text box]
- Transport: [Dropdown menu showing "TCP/IP"]
- Slow Connection
- OK button
- Cancel button

5. Next, enter the parameters.

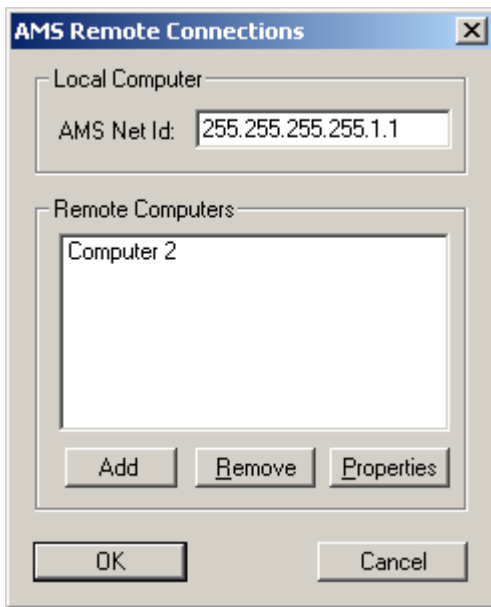
**Note:** This procedure must be repeated for all remote TwinCAT PLCs. The host computer must be added in the same way as the remote computers on each of the remote computers.



The screenshot shows the same "Add Remote Connection" dialog box, but with the following values entered:

- Name: Computer 2
- AMS Net Id: 255.255.255.1.1
- Address: 255.255.255.255
- Transport: TCP/IP
- Slow Connection
- OK button
- Cancel button

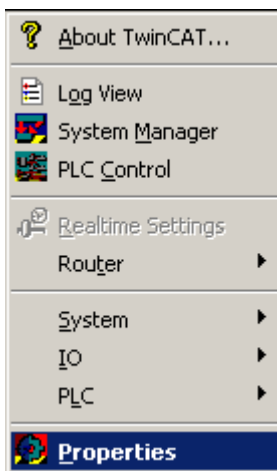
6. After all the settings have been entered, confirm by pressing **OK**.



### Configuration with TwinCAT PLC Installed

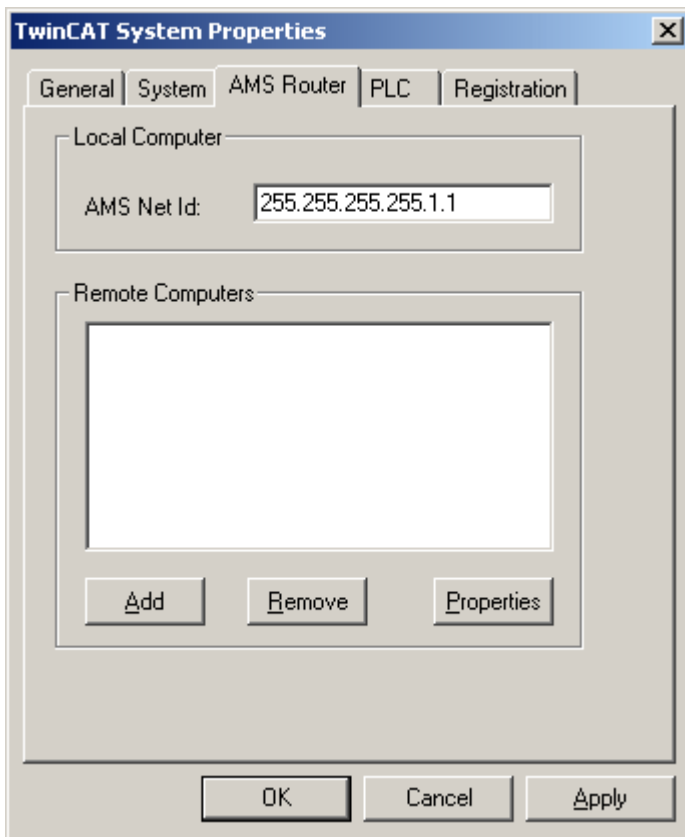
A TwinCAT-PC's ADS-AMS Net ID is set in the TwinCAT system service.

1. Click on the TwinCAT system service icon located within the notification area of the toolbar. Select **Properties** to open the **TwinCAT System Properties** dialog box.



2. Select the **AMS Router** tab.

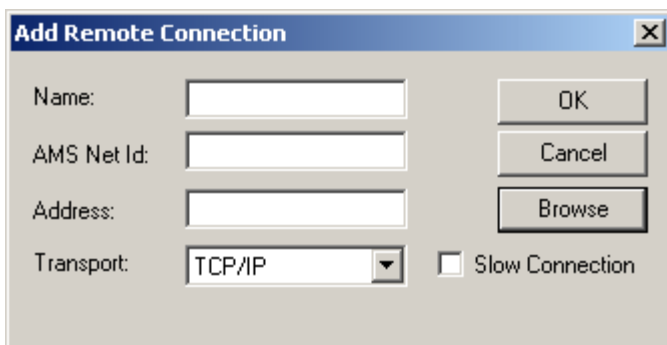
3. Read or modify the desired AMS Net Identifier in the **AMS Net ID** field. By default, the TwinCAT installation constructs the AMS Net ID from the TCP/IP address of the local PC with an extension of .1.1 as if it were a sub-net mask for field busses, target bus controllers or etc. The AMS Net ID can be freely chosen, however.



4. When an ADS device's services are called upon in the network, its **AMS Net ID** must be known. By inserting the target PC, TwinCAT can establish the connection between the TCP/IP address of the target PC and the AMS Net ID of the target message router address.

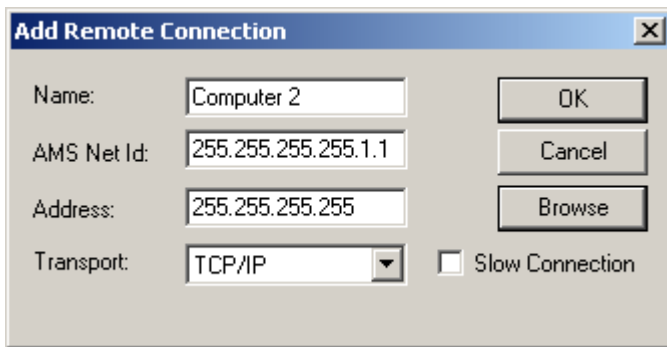
5. Similarly, in order to create an AMS connection between a multiple computers, the remote computers must also be known to each other. To do so, click **TwinCAT | Properties | AMS Router**. The **AMS Net ID** is composed of the TCP/IP of the local computer plus the suffix .1.1. The AMS Net ID is based on the TCP/IP address, but the relationship is not entirely fixed.

6. To include a remote computer in the connection, press **Add**.

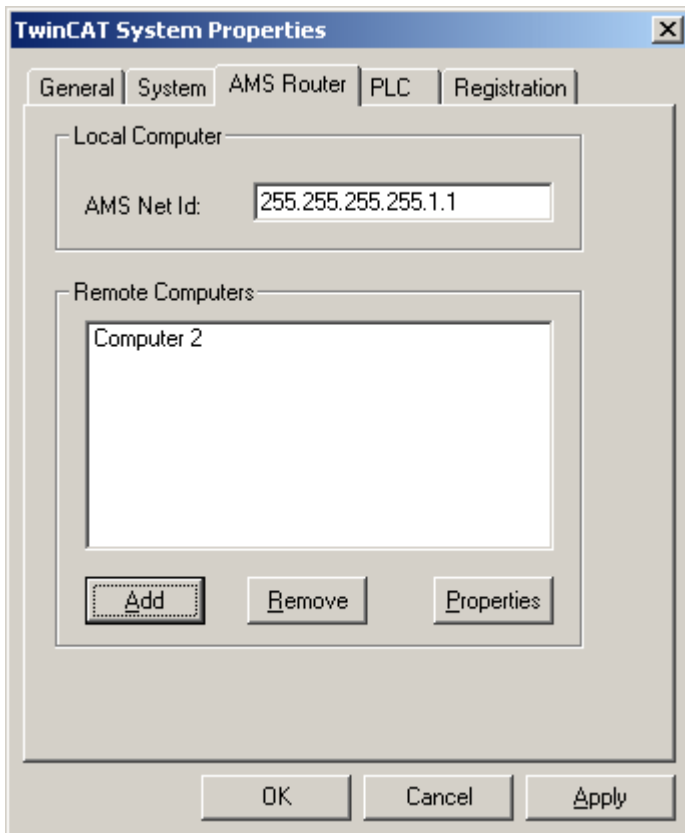


7. To add parameters, either enter manually or select by using **Browse** button.

**Note:** This procedure must be repeated for all remote computers. The host computer must be added in the same way as the remote computers on each of the remote computers.



8. To confirm the settings that have been entered, click **OK**.



9. The computers may now be restarted.

**Note:** For more information about the AMS messaging protocol, refer to Beckhoff's help documentation.

## Automatic Tag Database Generation

### Controller-to-Server Name Conversions

#### Leading underscores

Leading underscores ( \_ ) in tag/program names will be replaced with U\_ . This is required since the server does not accept tag/group names beginning with an underscore.

### Preparing for Automatic Tag Database Generation

#### Upload Symbols from Device

It is recommended that all communications to the Beckhoff TwinCAT of interest are halted during the database creation process.

### In the OPC Server

1. Open up **Device Properties** in the device for which tags will be generated.
2. Select the **Database Settings** tab.
3. Select **Create tag database from device**.
4. Select desired **Beckhoff Compatibility** options.
5. Select the **Options** tab and make desired changes.
6. Select the **Database Creation** tab and utilize as instructed in **Database Creation Settings**.

**Note:** At this time, tags and symbols can only be uploaded from a TwinCAT soft PLC. They cannot be uploaded from a hardware BC/BX controller.

### Upload Symbols from File

The Beckhoff TwinCAT driver uses a file generated from Beckhoff TwinCAT PLC called an .tpy import/export file to generate the tag database.

**Note:** All tags, including global and program, will be imported and expanded according to their respective data types.

### In Beckhoff TwinCAT PLC

The .tpy file is automatically created the first time that a build is done on a clean project.

1. Open the **TwinCAT PLC Control** program.
2. Click **Project | Clean All**.
3. Click **Project | Build**.

**Note 1:** The program will not generate the .tpy file again until the project has been cleaned. Clicking **Project | Rebuild All**, however, will always regenerate the .tpy file.

**Note 2:** The .tpy file is located in the same directory as the project's .pro file.

### In the OPC Server

1. Open up **Device Properties** in the device for which tags will be generated.
2. Select the **Database Settings** tab.
3. Select **Create tag database from import file**.
4. Enter or browse for the location of the **.tpy file** previously created.
5. Select desired **Beckhoff Compatibility** options.
6. Select the **Options** tab and make desired changes.
7. Select the **Database Creation** tab and utilize as instructed in **Database Creation Settings**.

**Note:** Online tag generation for PLC projects that contain **Step-Transitions** will produce Boolean tags for each step. The tag describes whether the state is active or inactive. Offline generation will not produce Step-Transition tags.

---

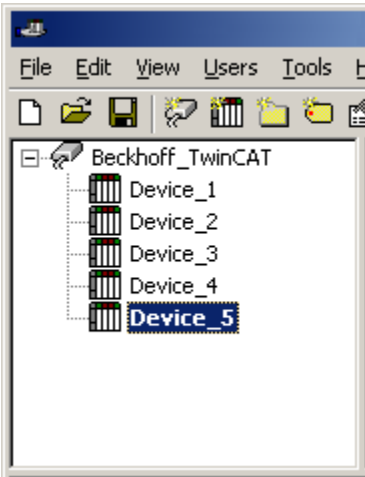
## Performance Optimization

### Optimizing Communications

With any programmable controller there are unique ways for optimizing system throughput, and the Beckhoff TwinCAT Driver is no different from the rest. The Beckhoff TwinCAT driver has been designed to optimize reads and writes. For tags of all data types, requests are grouped into a single transaction. This provides drastic improvement in performance over single tag transaction. The only limitation is on the number of tags that can fit in a single transaction.

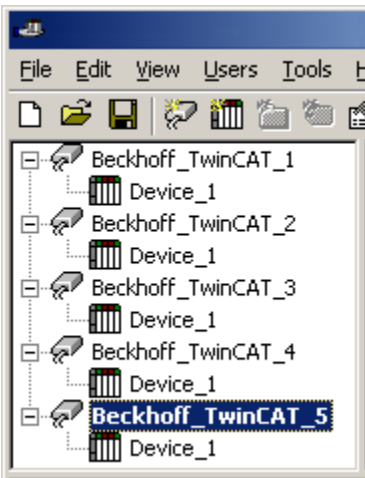
### Optimizing the Application

While the Beckhoff TwinCAT driver is fast, there are a couple of guidelines that can be used in order to control and optimize the application and gain maximum performance. Our server refers to communications protocols like Beckhoff TwinCAT as a channel. Each channel defined in the application represents a separate path of execution in the server. Once a channel has been defined, a series of devices must then be defined under that channel. Each of these devices represents a single Beckhoff TwinCAT controller from which data will be collected. While this approach to defining the application will provide a high level of performance, it won't take full advantage of the Beckhoff TwinCAT driver or the network. An example of how the application may appear when configured using a single channel is shown below.



In this example, each device appears under a single channel called Beckhoff\_TwinCAT. In this configuration, the driver must move from one device to the next as quickly as possible to gather information at an effective rate. As more devices are added or more information is requested from a single device, however, the overall update rate begins to suffer.

If the Beckhoff TwinCAT driver could only define one single channel, then the example shown above would be the only option available; however, the Beckhoff TwinCAT driver can define up to 100 channels. Using multiple channels distributes the data collection workload by simultaneously issuing multiple requests to the network. An example of how the same application may appear when configured using multiple channels to improve performance is shown below.



Each device has now been defined under its own channel. In this new configuration, a single path of execution is dedicated to the task of gathering data from each device. If the application has 100 or fewer devices, it can be optimized exactly how it is shown here.

The performance will improve even if the application has more than 100 devices. While 100 or fewer devices may be ideal, the application will still benefit from additional channels. Although by spreading the device load across all channels will cause the server to move from device to device again, it can now do so with far less devices to process on a single channel.

## Data Types Description

Data Type	Description
Boolean	Single bit
Byte	Unsigned 8 bit value
Char	Signed 8 bit value
Word	Unsigned 16 bit value
Short	Signed 16 bit value
DWord	Unsigned 32 bit value

Long	Signed 32 bit value
Float	32 bit IEEE Floating point
Double	64 bit IEEE Floating point
String	Null terminated character array

**Note:** For a description of TwinCAT-platform specific data types, refer to [TwinCAT Data Types](#).

### Unsupported Data Types

Unsupported data types include LBCD and BCD.

## Address Descriptions

The following table summarizes the valid addressing formats in the server.

Address Format	Notation	Example	Notes
Standard	<tag name>	tag_1	Tag cannot be an array
Array Element	<array tag name> [dim1, dim2, dim3]	tag_1 [2, 58, 547]	Dimension Range=1 to 3
String	<tag name>/<string length>	tag_1/4	Element Range=0 to 65535 Length Range=1 to 65535 The number of characters to read/write equals the string length.

**Note:** For information on how elements are referenced for 1, 2 and 3 dimensional arrays, refer to [Ordering of Array Data](#).

See Also: [Addressing Atomic Data Types](#)

## Symbolic Tag Based Addressing

The Beckhoff TwinCAT Driver uses a tag or symbol based addressing structure. These tags (commonly referred to as **Native Tags**) differ from conventional PLC data items in that the tag name itself is the address, not a physical or logical address.

### Client/Server Tag Address Rules

Beckhoff TwinCAT variable names correspond to Client/Server Tag addresses. Beckhoff TwinCAT variable names (entered via Beckhoff TwinCAT PLC) follow the IEC 61131-3 identifier rules. Client/Server Tag addresses follow these same rules and are listed below:

- Must begin with an alphabetic (A-Z, a-z) character or an underscore (\_).
- Can only contain alphanumeric characters and underscores.
- First 32 characters are significant.
- Can not have consecutive underscores.
- Are not case sensitive.

### Client/Server Tag Name Rules

The rules for tag name assignment in the server differ from address assignment because they cannot begin with an underscore.

## Tag Scope

Global tags are Beckhoff TwinCAT variables that have global scope in the controller. Any program or task can access the global tags, which use the following notation:

. <tag name>

### Program Tags

Program tags are identical to global tags, except that a program tag's scope is local to the program it is defined in. Program tags follow the same addressing rules and limitations as global tags. The only difference is that program tags are prefixed with the following notation:

**<program name> . <tag name>**

For example, Beckhoff TwinCAT variable tag\_1 in program prog\_1 would be addressed as prog\_1.tag\_1 in a Client/Server Tag address.

### Structure Tag Addressing

Beckhoff TwinCAT structure variables, global or program, contain one or more member variables. Member variables can be atomic or structured in nature.

Global tag example:	Program tag example:
.<structure name>.<atomic-type tag>	<program name>.<structure name>.<atomic-type tag>

This implies that a substructure would be addressed as:

Global tag example:	Program tag example:
.<structure name>.<substructure name>.<atomic-type tag>	<program name>.<structure name>.<substructure name>.<atomic-type tag>

Arrays of structures would be addressed as:

Global tag example:	Program tag example:
.<structure array name>[dim1, dim2, dim3].<atomic-type tag>	<program name>.<structure array name>[dim1, dim2, dim3].<atomic-type tag>

Again, this implies that an array of substructures would be addressed as:

Global tag example:	Program tag example:
.<structure name>.<substructure array name>[dim1, dim2, dim3].<atomic-type tag>	<program name>.<structure name>.<substructure array name>[dim1, dim2, dim3].<atomic-type tag>

**Note:** These are a few of the many addressing possibilities involving structures. These are shown here only to provide an introduction to structure addressing. For more information, refer to the Beckhoff TwinCAT documentation.

## Addressing Atomic Data Types

The table below shows suggested usages and addressing possibilities for a TwinCAT Data Type given the address formats available.

Atomic Data Type	Standard	Array Element	String
<b>BOOL</b>			
Client/Server Data Type	Boolean	Boolean	
Client Server Tag Example	BOOLTAG	BOOLARR[0]	
<b>SINT/USINT</b>			
Client/Server Data Type	Byte, Char	Byte, Char	String
Client/Server Tag Example	SINTTAG	SINTARR[0]	SINTARR[0]/4  String length cannot be longer than the array length minus the element position.

<b>INT/UINT</b>			
Client/Server Data Type	Word, Short	Word, Short	
Client/Server Tag Example	INTTAG	INTARR[0]	
<b>DINT/UDINT</b>			
Client/Server Data Type	DWord, Long	DWord, Long	
Client/Server Tag Example	DINTTAG	DINTARR[0]	
<b>REAL</b>			
Client/Server Data Type	Float	Float	
Client/Server Tag Example	REALTAG	REALARR[0]	
<b>LREAL*</b>			
Client/Server Data Type	Double	Double	
Client/Server Tag Example	LREALTAG	LREALARR[0]	
<b>TIME</b>			
Client/Server Data Type	DWord, Long	DWord, Long	
Client/Server Tag Example	TIMETAG	TIMEARR[0]	
<b>STRING</b>			
Client/Server Data Type	String	String	
Client/Server Tag Example	STRINGTAG/8 0	STRINGARR[1]/80	

\*This data type is not supported by the BC/BX Controller model.

### Unsupported TwinCAT Data Types

Unsupported data types include pointers, enumerations, TOD and DT.

See Also: [Address Formats](#)

## Ordering of TwinCat Array Data

### 1. Dimensional Arrays - array [dim1]

1 dimensional array data is passed to and from the controller in ascending order:

```
for (dim1=0; dim1<dim1_max; dim1++)
```

**Example:** 3 element array

```
array [0] array [1] array [2]
```

### 2. Dimensional Arrays - array [dim1, dim2]

2 dimensional array data is passed to and from the controller in ascending order:

```
for (dim1=0; dim1<dim1_max; dim1++)
for (dim2=0; dim2<dim2_max; dim2++)
```

**Example:** 3X3 element array

```
array [0, 0]
array [0, 1]
```

```
array [0, 2]
array [1, 0]
array [1, 1]
array [1, 2]
array [2, 0]
array [2, 1]
array [2, 2]
```

### 3. Dimensional Arrays - array [dim1, dim2, dim3]

3 dimensional array data is passed to and from the controller in ascending order:

```
for (dim1 = 0; dim1 < dim1_max; dim1++)
for (dim2 = 0; dim2 < dim2_max; dim2++)
for (dim3 = 0; dim3 < dim3_max; dim3++)
```

**Example:** 3X3x3 element array

```
array [0, 0, 0]
array [0, 0, 1]
array [0, 0, 2]
array [0, 1, 0]
array [0, 1, 1]
array [0, 1, 2]
array [0, 2, 0]
array [0, 2, 1]
array [0, 2, 2]
array [1, 0, 0]
array [1, 0, 1]
array [1, 0, 2]
array [1, 1, 0]
array [1, 1, 1]
array [1, 1, 2]
array [1, 2, 0]
array [1, 2, 1]
array [1, 2, 2]
array [2, 0, 0]
array [2, 0, 1]
array [2, 0, 2]
array [2, 1, 0]
array [2, 1, 1]
array [2, 1, 2]
array [2, 2, 0]
array [2, 2, 1]
array [2, 2, 2]
```

## Error Descriptions

---

The following error/warning messages may be generated. Click on the link for a description of the message.

### Address Validation

[Device Address '<address>' contains a syntax error](#)

[Data type '<type>' is not valid for device address '<address>'](#)

[Array size is out of range for address '<address>'](#)

[Address '<address>' is out of range for the specified device or register](#)

### Driver Error Messages

[Unable to load '<dll>'](#)

[Unable to import from '<dll>'](#)

[Port open failed with Error Code '<#>'](#)

### Device Status Messages

['Device <Device name>' is not responding](#)

[Unable to synchronize with configuration file '<file name>' on device '<device name>' due to memory allocation error](#)

[Unable to perform Beckhoff compatibility on device '<device name>' due to memory allocation error](#)  
[Unable to gather runtime information for device '<device name>'. Reason '<reason>'](#)

### Read Error Messages

[Unable to read tag '<address>' on device '<device name>'. Error Code '<#>'](#)  
[Unable to read tag '<address>' on device '<device name>'. Tag data size of '<size>' bytes\(s\) exceeds actual data size of '<size>' byte\(s\)](#)  
[Unable to read tag '<address>' on device '<device name>'. Address bounds exceeded](#)  
[Unable to read tag '<address>' on device '<device name>'. Memory allocation error](#)  
[Unable to read tag '<address>' on device '<device name>'. Actual data type is not compatible with tag of type '<data type>'](#)  
[Unable to read tag '<address>' on device '<device name>'. Runtime file is not valid](#)  
[Unable to read tag '<address>' on device '<device name>'. Symbol not found in file](#)  
[Unable to read tag '<address>' on device '<device name>'. Beckhoff TwinCAT DLL\(s\) necessary for runtime are not loaded](#)  
[Unable to read tag '<address>' on device '<device name>'. Tag does not meet filtering requirements](#)

### Write Error Messages

[Unable to write to tag '<address>' on device '<device name>'. Error Code '<#>'](#)  
[Unable to write to tag '<address>' on device '<device name>'. Tag data size of '<size>' bytes\(s\) exceeds actual data size of '<size>' byte\(s\)](#)  
[Unable to write to tag '<address>' on device '<device name>'. Address bounds exceeded](#)  
[Unable to write to tag '<address>' on device '<device name>'. Memory allocation error](#)  
[Unable to write to tag '<address>' on device '<device name>'. Actual data type is not compatible with tag of type '<data type>'](#)  
[Unable to write to tag '<address>' on device '<device name>'. Runtime file is not valid](#)  
[Unable to write to tag '<address>' on device '<device name>'. Symbol not found in file](#)  
[Unable to write to tag '<address>' on device '<device name>'. Beckhoff TwinCAT DLL\(s\) necessary for runtime are not loaded](#)  
[Unable to write to tag '<address>' on device '<device name>'. Tag does not meet filtering requirements](#)  
[Unable to write to tag '<address>' on device '<device name>'. Tag access is Read Only](#)

### Automatic Tag Database Generation Error Messages

[Unable to generate a tag database for device '<device name>'. Reason: Memory allocation error](#)  
[Unable to generate a tag database for device '<device name>'. Reason: Beckhoff TwinCAT DLL\(s\) necessary for import are not loaded](#)  
[Unable to generate a tag database for device '<device name>'. Reason: Device returned error code '<#>'](#)  
[Unable to generate a tag database for device '<device name>'. Reason: Device is not responding](#)

## Error Codes

The following sections define error codes that may be encountered in the event log of the server. Refer to the Event Log section within the Server Options chapter of the server help file for detailed information on how the event logger works.

Hex	Dec	Description
0x000	0	No error
0x001	1	Internal error
0x002	2	No Rtime
0x003	3	Allocation locked memory error
0x004	4	Insert mailbox error
0x005	5	Wrong receive HMSG
0x006	6	Target port not found
0x007	7	Target machine not found
0x008	8	Unknown command ID

0x009	9	Bad task ID
0x00A	10	No IO
0x00B	11	Unknown AMS command
0x00C	12	Win 32 error
0x00D	13	Port not connected
0x00E	14	Invalid AMS length
0x00F	15	Invalid AMS Net ID
0x010	16	Low Installation level
0x011	17	No debug available
0x012	18	Port disabled
0x013	19	Port already connected
0x014	20	AMS Sync Win32 error
0x015	21	AMS Sync Timeout
0x016	22	AMS Sync AMS error
0x017	23	AMS Sync no index map
0x018	24	Invalid AMS port
0x019	25	No memory
0x01A	26	TCP send error
0x01B	27	Host unreachable
0x500	1280	Router: no locked memory
0x502	1282	Router: mailbox full
0x700	1792	Error class <device error>
0x701	1793	Service is not supported by server
0x702	1794	Invalid index group
0x703	1795	Invalid index offset
0x704	1796	Reading/writing not permitted
0x705	1797	Parameter size not correct
0x706	1798	Invalid parameter value(s)
0x707	1799	Device is not in a ready state
0x708	1800	Device is busy
0x709	1801	Invalid context (must be in Windows)
0x70A	1802	Out of memory
0x70B	1803	Invalid parameter value(s)
0x70C	1804	Not found (files, ...)
0x70D	1805	Syntax error in command or file
0x70E	1806	Objects do not match
0x70F	1807	Object already exists
0x710	1808	Symbol not found
0x711	1809	Symbol version invalid
0x712	1810	Server is in invalid state
0x713	1811	AdsTransMode not supported
0x714	1812	Notification handle is invalid
0x715	1813	Notification client not registered
0x716	1814	No more notification handles
0x717	1815	Size for watch too big
0x718	1816	Device not initialized
0x719	1817	Device has a timeout
0x71A	1818	Query interface failed

0x71B	1819	Wrong interface required
0x71C	1820	Class ID is invalid
0x71D	1821	Object ID is invalid
0x71E	1822	Request is pending
0x71F	1823	Request is aborted
0x720	1824	Signal warning
0x721	1825	Invalid array index
0x740	1856	Error class <client error>
0x741	1857	Invalid parameter at service
0x742	1858	Polling list is empty
0x743	1859	Var connection already in use
0x744	1860	Invoke ID in use
0x745	1861	Timeout elapsed
0x746	1862	Error in win32 subsystem
0x748	1864	Ads-port not opened
0x750	1872	Internal error in ads sync
0x751	1873	Hash table overflow
0x752	1874	Key not found in hash
0x753	1875	No more symbols in cache

## Address Validation

---

The following error/warning messages may be generated. Click on the link for a description of the message.

### Address Validation

[Device Address '<address>' contains a syntax error](#)

[Data type '<type>' is not valid for device address '<address>'](#)

[Array size is out of range for address '<address>'](#)

[Address '<address>' is out of range for the specified device or register](#)

## Device Address '<address>' contains a syntax error

---

### Error Type:

Warning

### Possible Cause:

A tag address that has been specified statically contains one or more of the following errors:

1. The address doesn't conform to the tag address naming conventions.
2. The address is invalid according to the address format and underlying controller tag data type.
3. A program tag was specified incorrectly.
4. The address used an invalid format.

### Solution:

Re-enter the address in the client application.

## Data type '<type>' is not valid for device address '<address>'

---

### Error Type:

Warning

### Possible Cause:

A tag address that has been specified statically has been assigned an invalid data type.

**Solution:**

Modify the requested data type in the client application.

**Array size is out of range for address '<address>'**

---

**Error Type:**

Warning

**Possible Cause:**

A tag address that has been specified statically is requesting an array size that is too large.

**Solution:**

Re-enter the address in the client application to specify a smaller value for the array or a different starting point.

**Address '<address>' is out of range for the specified device or register**

---

**Error Type:**

Warning

**Possible Cause:**

A tag address that has been specified dynamically references a location that is beyond the range of supported locations for this device.

**Solution:**

Verify that the address is correct; if it is not, re-enter it in the client application.

**Driver Error Messages**

---

The following error/warning messages may be generated. Click on the link for a description of the message.

**Driver Error Messages**

[Unable to load '<dll>'](#)

[Unable to import from '<dll>'](#)

[Port open failed with Error Code '<#>'](#)

**Unable to load '<dll>'**

---

**Error Type:**

Serious

**Possible Cause:**

A software component necessary for communications with the Beckhoff TwinCAT AMS message router cannot be loaded from '<dll>'.

**Solution:**

Re-install the server and then try again.

**Unable to import from '<dll>'**

---

**Error Type:**

Serious

**Possible Cause:**

A software component necessary for communications with the Beckhoff TwinCAT AMS message router cannot be loaded from '<dll>'.

**Solution:**

Re-install the server and then try again.

### **Port open failed with Error Code '<#>'**

---

**Error Type:**

Serious

**Possible Cause:**

A software component necessary for communications with the Beckhoff TwinCAT AMS message router cannot be loaded from '<dll>'.  
</p></div>

**Solution:**

Re-install the server and then try again.

### **Device Status Error Messages**

---

The following error/warning messages may be generated. Click on the link for a description of the message.

**Device Status Messages**

[Device '<Device name>' is not responding](#)

[Unable to synchronize with configuration file '<file name>' on device '<device name>' due to memory allocation error](#)

[Unable to perform Beckhoff compatibility on device '<device name>' due to memory allocation error](#)

[Unable to gather runtime information for device '<device name>'. Reason '<reason>'](#)

### **Device '<Device name>' is not responding**

---

**Error Type:**

Warning

**Result:****If tag was being read:**

The Read operation will not be performed and the tag will be invalidated.

**If tag was being written:**

The Write operation for the given tag will not occur.

**Possible Cause:**

1. The connection between the device and the host PC is broken.
2. The named device may have been assigned an incorrect AMS Net ID address.
3. Device CPU work load is too high.

**Solution:**

1. Verify the cabling between the PC and the PLC device.
2. Verify that the AMS Net ID address given to the named device matches that of the actual device.
3. If this error occurs frequently, decrease the tag group scan rate to reduce the work load on the PLC CPU.
4. Increase the cycle time interval under Task Configurations in TwinCAT PLC Control.

### **Unable to synchronize with configuration file '<file name>' on device '<device name>' due to memory allocation error**

---

**Error Type:**

Warning

**Possible Cause:**

Memory required to synchronizing with .tpy runtime file could not be allocated. The original uploaded file will continue to be used.

**Solution:**

Close any unused applications and/or increase the amount of virtual memory. Then, restart the server and try again.

**Unable to perform Beckhoff compatibility on device '<device name>' due to memory allocation error**

---

**Error Type:**

Warning

**Possible Cause:**

The memory required for device tags to meet Beckhoff compatibility could not be allocated.

**Solution:**

Close any unused applications and/or increase the amount of virtual memory. Then, restart the server and try again.

**Unable to gather runtime information for device '<device name>'. Reason '<reason>'**

---

**Error Type:**

Serious

**Possible Cause:**

The cause depends on the error reason returned.

**Solution:**

The solution depends on the error reason returned.

**See Also:**

[Error Codes](#)

**Read Error Messages**

---

The following error/warning messages may be generated. Click on the link for a description of the message.

**Read Error Messages**

[Unable to read tag '<address>' on device '<device name>'. Error Code '<#>'](#)

[Unable to read tag '<address>' on device '<device name>'. Tag data size of '<size>' bytes\(s\) exceeds actual data size of '<size>' byte\(s\)](#)

[Unable to read tag '<address>' on device '<device name>'. Address bounds exceeded](#)

[Unable to read tag '<address>' on device '<device name>'. Memory allocation error](#)

[Unable to read tag '<address>' on device '<device name>'. Actual data type is not compatible with tag of type '<data type>'](#)

[Unable to read tag '<address>' on device '<device name>'. Runtime file is not valid](#)

[Unable to read tag '<address>' on device '<device name>'. Symbol not found in file](#)

[Unable to read tag '<address>' on device '<device name>'. Beckhoff TwinCAT DLL\(s\) necessary for runtime are not loaded](#)

[Unable to read tag '<address>' on device '<device name>'. Tag does not meet filtering requirements](#)

**Unable to read tag '<address>' on device '<device name>'. Error Code '<#>'**

---

**Error Type:**

Warning

**Possible Cause:**

The cause depends on the error code(s) returned.

**Solution:**

The solution depends on the error code(s) returned.

**See Also:**

[Error Codes](#)

---

**Unable to read tag '<address>' on device '<device name>'. Tag data size of '<size>' bytes(s) exceeds actual data size of '<size>' byte(s)**

---

**Error Type:**

Warning

**Possible Cause:**

A read request for the specified tag failed because the controller's tag data type is smaller than the requested data type.

**Solution:**

Change the tag's data type size to a size that is smaller than or equal to the ones available on the controller.

---

**Unable to read tag '<address>' on device '<device name>'. Address bounds exceeded**

---

**Error Type:**

Warning

**Possible Cause:**

The controller's tag size is smaller than the requested data size.

**Solution:**

Ensure that the requested data size matches the address size in the controller.

---

**Unable to read tag '<address>' on device '<device name>'. Memory allocation error**

---

**Error Type:**

Warning

**Possible Cause:**

The memory required for the read request could not be allocated.

**Solution:**

Close any unused applications and/or increase the amount of virtual memory. Then, restart the server and try again.

---

**Unable to read tag '<address>' on device '<device name>'. Actual data type is not compatible with tag of type '<data type>'**

---

**Error Type:**

Warning

**Possible Cause:**

1. A read request for tag '<address>' is unable to be cast.
2. A read request for tag '<address>' corresponds to an unsupported data type.

**Solution:**

1. Change the tag's data type to one that is supported.
2. Contact Technical Support so that support can be added for this type.

**Note:**

The data type of a item in the server should match the data type for the controller address that it references.

**Unable to read tag '<address>' on device '<device name>'. Runtime file is not valid****Error Type:**

Warning

**Possible Cause:**

1. There is no file loaded into the project.
2. The file loaded does not have a valid \*.tpy extension.
3. The file has been modified outside of TwinCAT PLC Control.

**Solution:**

1. Ensure that there is a valid .tpy file loaded into the project.
2. In TwinCAT PLC Control, rebuild the project and load the new .tpy file into the project.

**Unable to read tag '<address>' on device '<device name>'. Symbol not found in file****Error Type:**

Warning

**Possible Cause:**

The tag '<address>' cannot be found in the .tpy file.

**Solution:**

1. Ensure that the correct .tpy file is loaded into the project .
2. Ensure that the correct tag '<address>' is entered.

**Unable to read tag '<address>' on device '<device name>'. Beckhoff TwinCAT DLL(s) necessary for runtime are not loaded****Error Type:**

Serious

**Possible Cause:**

A software component necessary for communication with the Beckhoff TwinCAT AMS message router cannot be loaded from '<dll>'.

**Solution:**

Re-install the server and then try again.

**Unable to read tag '<address>' on device '<device name>'. Tag does not meet filtering requirements****Error Type:**

Warning

**Possible Cause:**

A read request on tag '<address>' does not meet the filtering requirements defined on the controller.

1. The tag does not have filtering properties.
2. The filter string entered does not match the string defined on the controller.

**Solution:**

Ensure that the controller is filtering the tag. Then, correct the filter string spelling.

**Write Error Messages**

The following error/warning messages may be generated. Click on the link for a description of the message.

## Write Error Messages

[Unable to write to tag '<address>' on device '<device name>'. Error Code '<#>'](#)

[Unable to write to tag '<address>' on device '<device name>'. Tag data size of '<size>' bytes\(s\) exceeds actual data size of '<size>' byte\(s\)](#)

[Unable to write to tag '<address>' on device '<device name>'. Address bounds exceeded](#)

[Unable to write to tag '<address>' on device '<device name>'. Memory allocation error](#)

[Unable to write to tag '<address>' on device '<device name>'. Actual data type is not compatible with tag of type '<data type>'](#)

[Unable to write to tag '<address>' on device '<device name>'. Runtime file is not valid](#)

[Unable to write to tag '<address>' on device '<device name>'. Symbol not found in file](#)

[Unable to write to tag '<address>' on device '<device name>'. Beckhoff TwinCAT DLL\(s\) necessary for runtime are not loaded](#)

[Unable to write to tag '<address>' on device '<device name>'. Tag does not meet filtering requirements](#)

[Unable to write to tag '<address>' on device '<device name>'. Tag access is Read Only](#)

---

## Unable to write to tag '<address>' on device '<device name>'. Error Code '<#>'

### Error Type:

Warning

### Possible Cause:

The cause depends on the error code(s) returned.

### Solution:

The solution depends on the error code(s) returned.

### See Also:

[Error Codes](#)

---

## Unable to write to tag '<address>' on device '<device name>'. Tag data size of '<size>' bytes(s) exceeds actual data size of '<size>' byte(s)

### Error Type:

Warning

### Possible Cause:

A write request for the specified tag failed because the controller's tag data type is smaller than the requested data type.

### Solution:

Change the tag's data type size to a size that is smaller than or equal to the ones available on the controller.

---

## Unable to write to tag '<address>' on device '<device name>'. Address bounds exceeded

### Error Type:

Warning

### Possible Cause:

The controller's tag size is smaller than the requested data size.

### Solution:

Ensure that the requested data size matches the address size in the controller.

**Unable to write to tag '<address>' on device '<device name>'. Memory allocation error****Error Type:**

Warning

**Possible Cause:**

The memory required for the write request could not be allocated.

**Solution:**

Close any unused applications and/or increase the amount of virtual memory. Then, restart the server and try again.

**Unable to write to tag '<address>' on device '<device name>'. Actual data type is not compatible with tag of type '<data type>'****Error Type:**

Warning

**Possible Cause:**

1. A write request for tag '<address>' is unable to be cast.
2. A write request for tag '<address>' corresponds to an unsupported data type.

**Solution:**

1. Change the tag's data type to one that is supported.
2. Contact Technical Support so that support can be added for this type.

**Note:**

The data type of a item in the server should match the data type for the controller address that it references.

**Unable to write to tag '<address>' on device '<device name>'. Runtime file is not valid****Error Type:**

Warning

**Possible Cause:**

1. There is no file loaded into the project.
2. The file loaded does not have a valid \*.tpy extension.
3. The file has been modified outside of TwinCAT PLC Control.

**Solution:**

1. Ensure that there is a valid .tpy file loaded into the project.
2. In TwinCAT PLC Control, rebuild the project and load the new .tpy file into the project.

**Unable to write to tag '<address>' on device '<device name>'. Symbol not found in file****Error Type:**

Warning

**Possible Cause:**

The tag '<address>' cannot be found in the .tpy file.

**Solution:**

1. Ensure that the correct .tpy file is loaded into the project .
2. Ensure that the correct tag '<address>' is entered.

**Unable to write to tag '<address>' on device '<device name>'. Beckhoff TwinCAT DLL (s) necessary for runtime are not loaded****Error Type:**

Serious

**Possible Cause:**

A software component necessary for communication with the Beckhoff TwinCAT AMS message router cannot be loaded from '<dll>'.

**Solution:**

Re-install the server and then try again.

**Unable to write to tag '<address>' on device '<device name>'. Tag does not meet filtering requirements**

---

**Error Type:**

Warning

**Possible Cause:**

A write request on tag '<address>' does not meet the filtering requirements defined on the controller.

1. The tag does not have filtering properties.
2. The filter string entered does not match the string defined on the controller.

**Solution:**

Ensure that the controller is filtering the tag. Then, correct the filter string spelling.

**Unable to write to tag '<address>' on device '<device name>'. Tag access is Read Only**

---

**Error Type:**

Warning

**Possible Cause:**

The write request on tag '<address>' does not meet OPC properties as defined on the controller.

**Solution:**

Change the tag properties defined on the controller to Read/Write access.

**Automatic Tag Database Generation Error Messages**

---

The following error/warning messages may be generated. Click on the link for a description of the message.

**Automatic Tag Database Generation Error Messages**

[Unable to generate a tag database for device '<device name>'. Reason: Memory allocation error](#)

[Unable to generate a tag database for device '<device name>'. Reason: Beckhoff TwinCAT DLL\(s\) necessary for import are not loaded](#)

[Unable to generate a tag database for device '<device name>'. Reason: Device returned error code '<#>'](#)

[Unable to generate a tag database for device '<device name>'. Reason: Device is not responding](#)

**Unable to generate a tag database for device '<device name>'. Reason: Memory allocation error**

---

**Error Type:**

Warning

**Possible Cause:**

The memory required for database generation could not be allocated. The process was aborted.

**Solution:**

Close any unused applications and/or increase the amount of virtual memory. Then, try again.

**Unable to generate a tag database for device '<device name>'. Reason: Beckhoff TwinCAT DLL(s) necessary for import are not loaded**

---

**Error Type:**

Warning

**Possible Cause:**

A software component necessary for communications with the Beckhoff TwinCAT AMS message router for database generation cannot be loaded from '<dll>'. The process has been aborted.

**Solution:**

Re-install the server and then try again.

**Unable to generate a tag database for device '<device name>'. Reason: Device returned error code '<#>'**

---

**Error Type:**

Warning

**Possible Cause:**

The cause depends on the error code(s) returned. The process has been aborted.

**Solution:**

The solution depends on the error code(s) returned.

**See Also:**

[Error Codes](#)

**Unable to generate a tag database for device '<device name>'. Reason: Device is not responding**

---

**Error Type:**

Warning

**Possible Cause:**

1. The connection between the device and the host PC is broken.
2. The named device may have been assigned an incorrect AMS Net ID address.
3. Device CPU work load is too high and has caused the process to be aborted.

**Solution:**

1. Verify the cabling between the PC and the PLC device.
2. Verify that the AMS Net ID address given to the named device matches that of the actual device.
3. If this error occurs frequently, decrease the tag group scan rate to reduce the work load on the PLC CPU.
4. Increase the cycle time interval under Task Configurations in TwinCAT PLC Control.

**Technical Notes**

---

For more information on specific aspects of the Beckhoff TwinCAT PLC, refer to the Technical Notes listed below.

**[TwinCAT Memory Warning](#)**

This notice discusses downloading a project while symbol handles are declared statically.

**[TwinCAT Time Slice Notes](#)**

This notice discusses the TwinCAT PLC's cyclic interval time.

## **TwinCAT Memory Warning**

---

Users should note the effects that declaring symbol handles statically and using Automatic Tag Database Generation have on the Beckhoff TwinCAT project. If the project is too large, the router will not have enough memory to perform this operation. To resolve this issue, refer to and use one of the solutions described below.

### **Solution 1: Change the Handles from Static to Dynamic**

1. Open the **TwinCAT PLC application**.
2. Click on the **Resources** tab.
3. Double-click on **Workspace**.
4. In the new window, select **TwinCAT**.
5. The window on the right should now contain a **Symbol Download** box. Select **Dynamic Symbols**.

### **Solution 2: Change the Router Memory Size**

1. Open **TwinCAT System Manager**.
2. Expand **SYSTEM-Configuration**. Then, click on **Real-Time Settings**.
3. Click on the **Settings** tab.
4. Increase the **Router Memory** size.

### **Solution 3: Upload the Project Offline from a .TPY File**

## **TwinCAT Time Slice Notes**

---

For proper operation, there must be enough time allocated between cyclic intervals. If the time interval is too small, performance will diminish significantly. Negative effects include the following:

- More time needed to perform online Automatic Tag Database Generation.
- Slower update rates in the OPC Client.

**Note:** The router memory size is set to 2048KB by default. The size can be increased to a maximum of 32768KB.

### **Increasing the Cyclic Interval Time**

Follow the instructions below for information on increasing the cyclic interval time.

1. Open the **TwinCAT PLC Control**.
2. Click on the **Resources** tab.
3. Double-click on **Task Configuration**, and then expand **Task Configuration** at the top of the middle window.
4. Select **Standard**.
5. In the **Task Attributes** tab, change the **Interval** value within the **Properties** box.

# Index

## - A -

Address '<address>' is out of range for the specified device or register 21  
 Address Formats 14  
 Address Validation 20  
 Addressing Atomic Data Types 15  
 Automatic Tag Database Generation 11  
 Automatic Tag Database Generation Error Messages 28

## - D -

Data type '<type>' is not valid for device address '<address>'. 20  
 Data Types Description 13  
 Device '<Device name>' is not responding 22  
 Device Address '<address>' contains a syntax error. 20  
 Device Setup 11  
 Device Status Error Messages 22  
 Driver Error Messages 21

## - E -

Error Codes 18  
 Error Descriptions 17

## - H -

Help Contents 3

## - O -

Overview 3

## - P -

Performance Optimization 12  
 Port open failed with Error Code '<#>' 22

## - R -

Read Error Messages 23

## - S -

Setting Up an ADS Remote Connection 7

## - T -

Tag Scope 14  
 Technical Notes 29  
 TwinCAT Memory Warning 30  
 TwinCAT Time Slice Notes 30

## - U -

Unable to gather runtime information for device '<device name>'. Reason '<reason>' 23  
 Unable to generate a tag database for device '<device name>'. Reason: Beckhoff TwinCAT DLL(s) necessary for import are not loaded 29  
 Unable to generate a tag database for device '<device name>'. Reason: Device is not responding 29  
 Unable to generate a tag database for device '<device name>'. Reason: Device returned error code '<#>' 29  
 Unable to generate a tag database for device '<device name>'. Reason: Memory allocation error 28  
 Unable to import from '<dll>' 21  
 Unable to load '<dll>'. 21  
 Unable to perform Beckhoff compatibility on device '<device name>' due to memory allocation error 23  
 Unable to read tag '<address>' on device '<device name>'. Actual data type is not compatible with tag of type '<data type>' 24  
 Unable to read tag '<address>' on device '<device name>'. Address bounds exceeded 24  
 Unable to read tag '<address>' on device '<device name>'. Error Code '<#>' 23  
 Unable to read tag '<address>' on device '<device name>'. Memory allocation error 24  
 Unable to read tag '<address>' on device '<device name>'. Tag data size of '<size>' bytes(s) exceeds actual data size of '<size>' byte(s) 24

Unable to read tag '<address>' on device '<device name>'. Beckhoff TwinCAT DLL(s) necessary for runtime are not loaded 25

Unable to read tag '<address>' on device '<device name>'. Runtime file is not valid 25

Unable to read tag '<address>' on device '<device name>'. Symbol not found in file 25

Unable to read tag '<address>' on device '<device name>'. Tag does not meet filtering requirements 25

Unable to synchronize with configuration file '<file name>' on device '<device name>' due to memory allocation error 22

Unable to write to tag '<address>' on device '<device name>'. Actual data type is not compatible with tag of type '<data type>' 27

Unable to write to tag '<address>' on device '<device name>'. Address bounds exceeded 26

Unable to write to tag '<address>' on device '<device name>'. Error Code '<#>' 26

Unable to write to tag '<address>' on device '<device name>'. Memory allocation error 27

Unable to write to tag '<address>' on device '<device name>'. Tag data size of '<size>' bytes(s) exceeds actual data size of '<size>' byte(s) 26

Unable to write to tag '<address>' on device '<device name>'. Beckhoff TwinCAT DLL(s) necessary for runtime are not loaded 27

Unable to write to tag '<address>' on device '<device name>'. Runtime file is not valid 27

Unable to write to tag '<address>' on device '<device name>'. Symbol not found in file 27

Unable to write to tag '<address>' on device '<device name>'. Tag access is Read Only 28

Unable to write to tag '<address>' on device '<device name>'. Tag does not meet filtering requirements 28

## - W -

Write Error Messages 25