Beckhoff TwinCAT Driver Help

© 2012 Kepware Technologies

Table of Contents

Beckhoff TwinCAT Driver Help

Help version 1.049

CONTENTS

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 the following:

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

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 to also exchange data with all registered TwinCAT servers worldwide. The message router 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 Runtime system. Under the CE operating system and the embedded operating systems for the series BX and BC controllers, only TwinCAT Runtime is available. Program modifications are implemented via network-capable powerful communication with the Runtime 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 per channel is 1024.

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 valid range is 100 to 30000 milliseconds. The default setting is 1000 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 valid range is 1 to 10. The default setting is 3 retries.

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
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 TwinCAT 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**.

Important: On Windows Vista and above, User Account Control (UAC) must be turned off before a remote connection may be configured with the Beckhoff TwinCAT Remote Manager Utility. If the remote connection is created before UAC is disabled, the configuration settings will be created in an incorrect location (resulting in poor performance).

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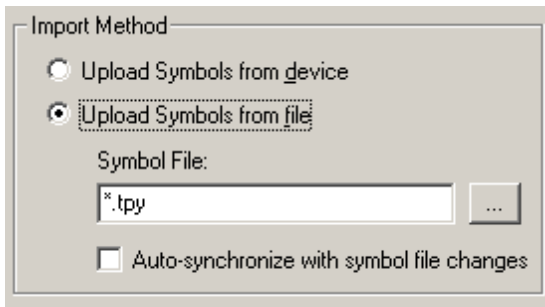This parameter specifies the data type that will be assigned to a Client/Server tag when the default type is selected during tag addition, modification, and import. Client/Server tags are assigned the default data type when any of the following conditions occur:

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.

```
┌─ Import Method ─────────────────────────────┐
│  ○ Upload Symbols from device               │
│  ⊙ Upload Symbols from file                 │
│                                             │
│      Symbol File:                           │
│      ┌──────────────────────────┐  ┌─────┐  │
│      │ *.tpy                    │  │ ... │  │
│      └──────────────────────────┘  └─────┘  │
│                                             │
│      ☐ Auto-synchronize with symbol file changes │
└─────────────────────────────────────────────┘
```

Description of the parameters are as follows:

- **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: The BC9xxx and BX9xxx models only 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.

```
┌─ Import and Runtime Compatibility ──────────┐
│                                             │
│   ☐ Respect OPC read/write item properties  │
│   ☐ Only import variables marked for OPC    │
│                                             │
│   Filter: ┌──────────────────────────────┐  │
│           │                              │  │
│           └──────────────────────────────┘  │
│                                             │
└─────────────────────────────────────────────┘
```
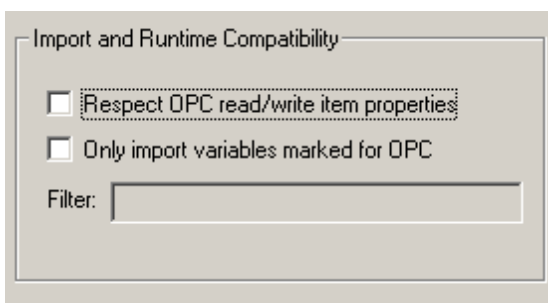
Respect OPC Read/Write Item Properties
When enabled, variables that include the string "OPC_PROP[0005]: <r/w access number>" (defined within the variable comment field in the PLC project) will be assigned the appropriate Read/Write access. During import and runtime, any tag(s) corresponding to a variable 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 run-time. 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

Important: On Windows Vista and above, User Account Control (UAC) must be turned off before a remote connection may be configured with the Beckhoff TwinCAT Remote Manager Utility. If the remote connection is created before UAC is disabled, the configuration settings will be created in an incorrect location (resulting in poor performance).

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. Next, double-click on **TcAmsRemoteMgr.exe**.

3. Beckhoff TwinCAT constructs the AMS Net ID from the TCP/IP address of the local PC by default, and uses an extension of ".1.1". The AMS Net ID can, however, be chosen freely.
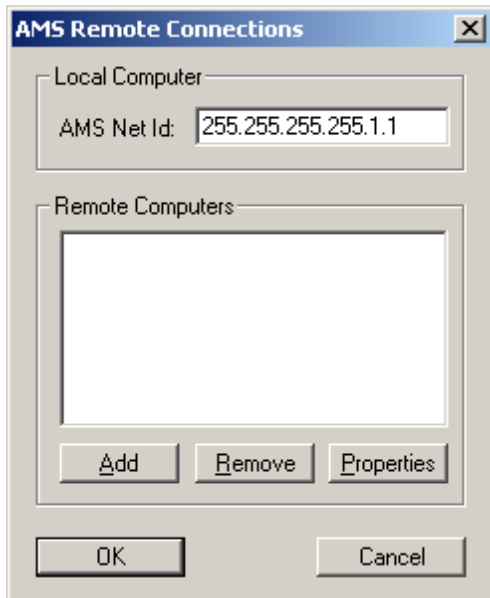
4. Next, press **Add** to include a remote computer in the connection.
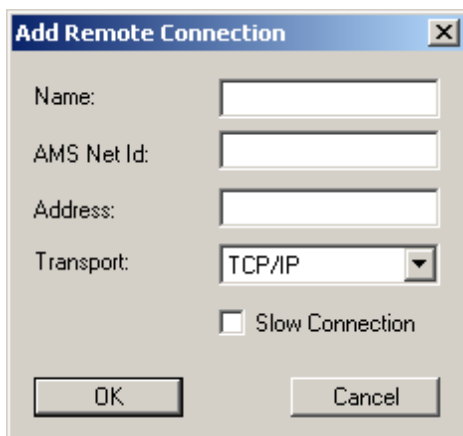
5. Then, specify the parameters.

   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.
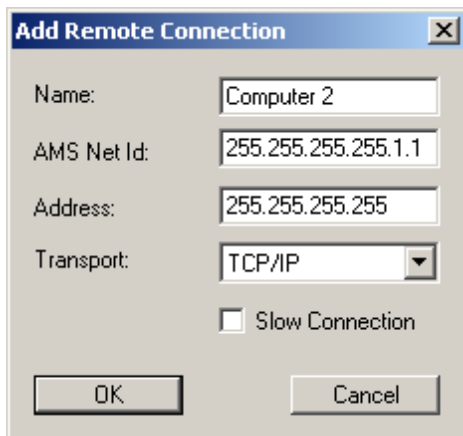
6. Once all the settings have been entered, press **OK**.
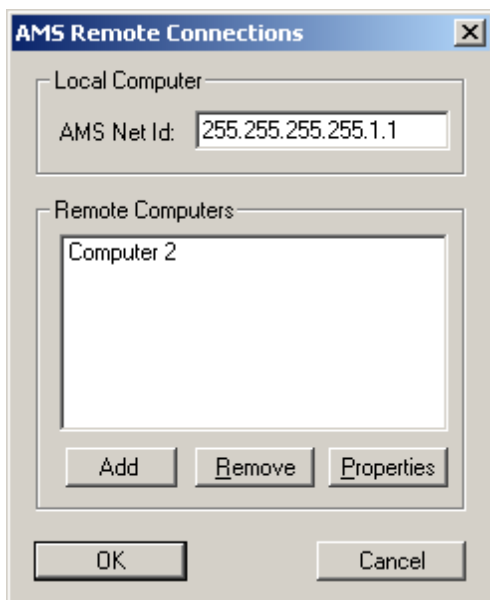
Configuration with TwinCAT PLC Installed

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

1. To start, click on the TwinCAT system service icon located within the toolbar's notification area. Then, select **Properties**.
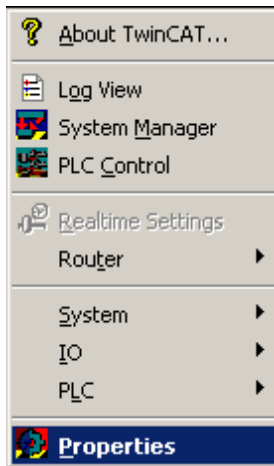
2. In **TwinCAT System Properties**, select the **AMS Router** tab.
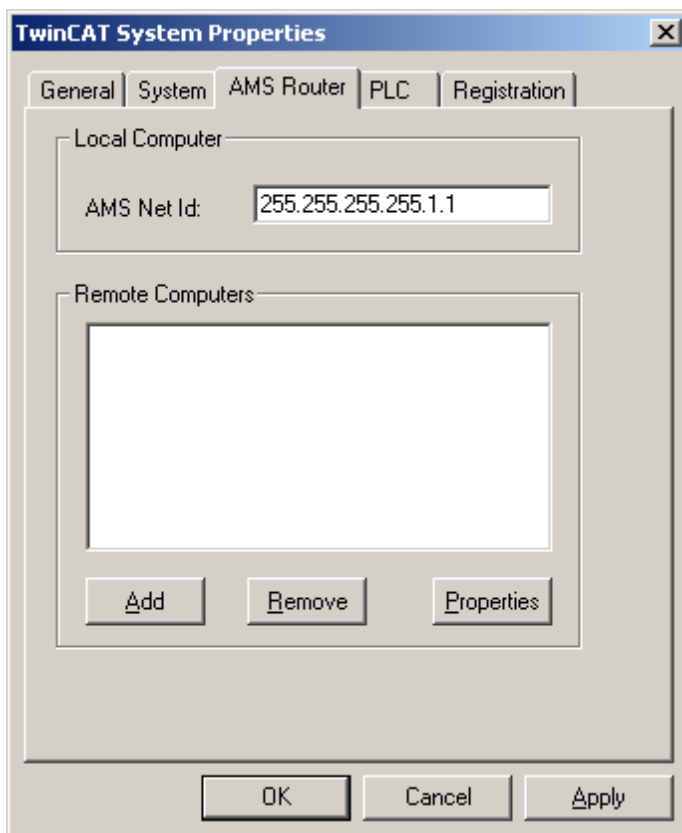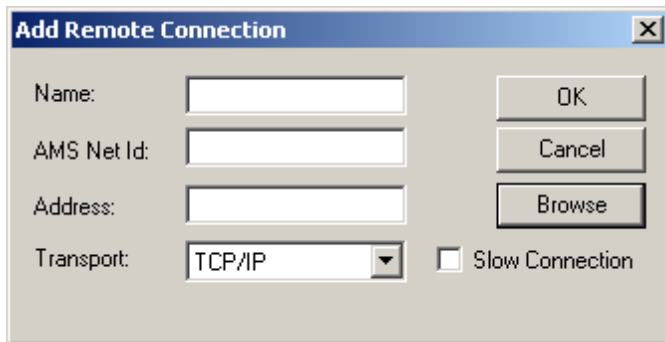
3. In **AMS Net ID**, read or modify the desired AMS Net Identifier. The TwinCAT installation constructs the AMS Net ID from the TCP/IP address of the local PC by default, and uses an extension of .1.1 (as if it were a sub-net mask for field buses, target bus controllers, and so forth). The AMS Net ID can, however, be freely chosen.

   Note: 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.

4. The remote computers must be known to each other in order to create an AMS connection between a multiple computers. 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.

5. Next, press **Add** to include a remote computer in the connection.
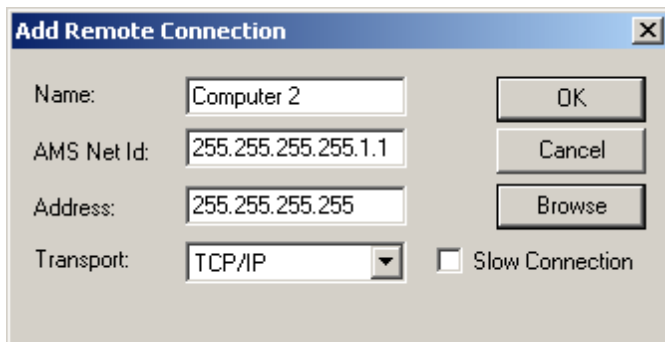
6. Then, either specify the parameters manually or select by clicking **Browse**.

   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.

7. Once all the settings have been entered, press **OK**.
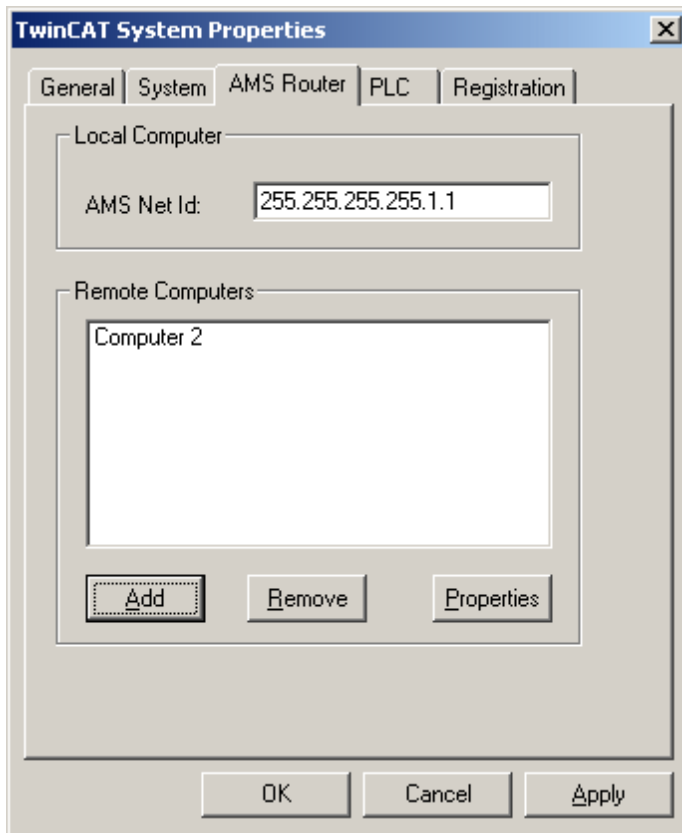
8. Restart the computers.

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. To start, open the **Device Properties** for the device in which tags will be generated.

2. Select the **Database Settings** tab, and then select **Create tag database from device**.

3. Next, select the desired **Beckhoff Compatibility** options.

4. Then, select the **Options** tab and make the desired changes.

5. 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. Then, 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. To start, open the **Device Properties** for the device in which tags will be generated.

2. Select the **Database Settings** tab, and then select **Create tag database from import file**.

3. Enter or browse for the location of the .tpy file that was previously created.

4. Next, select the desired **Beckhoff Compatibility** options.

5. Then, select the **Options** tab and make the desired changes.

6. 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.
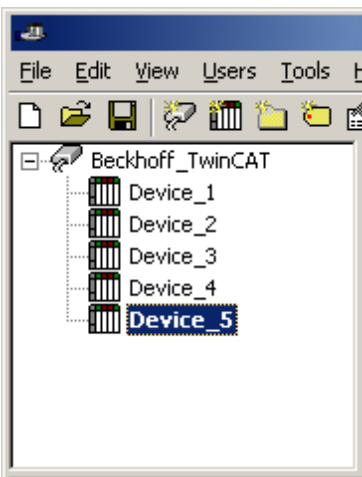
In this example, each device appears under a single channel called Beckhoff_TwinCAT. In this configuration, the driver must move from one device to the next as quickly as possible to gather information at an effective rate. As more devices are added or more information is requested from a single device, 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.
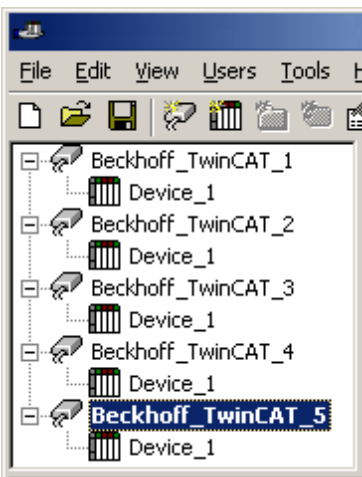
Each device has now been defined under its own channel. In this new configuration, a single path of execution is dedicated to the task of gathering data from each device. If the application has 100 or fewer devices, it can be optimized exactly how it is shown here.

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.

Adding SYSTEMINFO Global Variables to a TwinCAT PLC Project

Because the PLC system global variable "onlineChangeCount" is disabled by default, the PLC project must be changed in order to include it. The variable is added through the correct PLC library, which depends on both the

version of TwinCAT PLC Control and the target PLC system type (such as PC controller, BC controller, BX controller, and so forth) being used.

Including the "onlineChangeCount" variable in a PLC will allow the server project to access PLC variables using tag handles. Without it, the server must use fully qualified Symbolic information strings in each tag read. The number of bytes in a Symbolic information string can be several times larger than the number of bytes in a tag handle. For more information on adding the "onlineChangeCount" variable, refer to the instructions below.

Note: The following example applies to TwinCAT development environment version 2.11, and describes how to include the SYSTEMINFO variables in a TwinCAT PLC project where the target system is a PC controller. It assumes there is a pre-existing PLC project.

1. In **TwinCAT PLC Control**, load the PLC project. Do not login.

2. Next, locate and open the **Resources** tab at the bottom of the **Object Organizer**. Then, expand the **Resources** tree and double-click on **Library Manager**.

    Note: The Library Manager should be displayed in the TwinCAT PLC Control's work area. All libraries that are attached to a PLC project will be listed in the top left of the Library Manager. New projects should only have the "STANDARD.LIB" library attached.

3. To add another library, right-click in the open area beneath "STANDARD.LIB" and select **Additional Library**. Then, locate and double-click on **PlcSystem.lib**.

    Note: The top left of the Library Manager should now display "PlcSystem.lib".

4. Return to the **Library Manager**. Then, select **PlcSystem.lib** and open the **Data Types** tab.

5. Next, open the **SystemDataTypes** folder and select the **SYSTEMINFOTYPE** structure. The variable "onlineChangeCount" should be included within the Library Manager.

6. Login to the PLC project.

7. In the **Resources** tree, open the **Global Variables** folder. Then, double-click on the **Global_Variables** icon. At this point, the variable "onlineChangeCount" should appear as an element of structure "SystemInfo" in the work area.

    Important: The "SystemInfo" structure will only be displayed within "Global_Variables" after the user has logged in to the TwinCAT PLC Control.

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] tag_1 [0,3]	Dimension Range=1 to 3 Element Range=0 to 65535
String	<tag name>/<string length>	tag_1/4	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 (which are 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.
- Cannot have consecutive underscores.
- Are not case sensitive.

Client/Server Tag Name Rules

The rules for tag name assignment in the server differs from address assignment because tag names 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:
.<stucture name>.<atomic-type tag>	<program name>.<stucture 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	Client/Server	Standard	Array Element	String
BOOL	Data Type Tag Example	Boolean BOOLTAG	Boolean BOOLARR[0]	
SINT/USINT	Data Type Tag Example	Byte, Char SINTTAG	Byte, Char SINTARR[0]	String SINTARR[0]/4 String length cannot be longer then the array length minus the element position.
INT/UINT	Data Type Tag Example	Word, Short INTTAG	Word, Short INTARR[0]	
DINT/UDINT	Data Type Tag Example	DWord, Long DINTTAG	DWord, Long DINTARR[0]	
REAL	Data Type Tag Example	Float REALTAG	Float REALARR[0]	
LREAL*	Data Type Tag Example	Double LREALTAG	Double LREALARR[0]	
TIME	Data Type Tag Example	DWord, Long TIMETAG	DWord, Long TIMEARR[0]	
STRING	Data Type Tag Example	String STRINGTAG/80	String 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 Descriptions

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
Address '<address>' is out of range for the specified device or register
Array size is out of range for address '<address>'
Data type '<type>' is not valid for device address '<address>'
Device Address '<address>' contains a syntax error

Driver Error Messages
Port open failed with Error Code '<#>'
Unable to import from '<dll>'
Unable to load '<dll>'

Device Status Messages
Device '<Device name>' is not responding
Monitoring the global variable '<tag address>' on device '<device name>' in order to update symbol information when a change is detected
Unable to gather runtime information for device '<device name>'. Reason '<reason>'
Unable to perform Beckhoff compatibility on device '<device name>' due to memory allocation error
Unable to synchronize with configuration file '<file name>' on device '<device name>' due to memory allocation error
Unable to read tag '<tag address>' on device '<device name>'. Requesting symbol information for each transaction

Read Error Messages
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>'. Address bounds exceeded
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>'. Error Code '<#>'
Unable to read tag '<address>' on device '<device name>'. Memory allocation error
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>'. Tag data size of '<size>' bytes(s) exceeds actual data size of '<size>' byte(s)
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>'. Actual data type is not compatible with tag of type '<data type>'
Unable to write to tag '<address>' on device '<device name>'. Address bounds exceeded
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>'. Error Code '<#>'
Unable to write to tag '<address>' on device '<device name>'. Memory allocation error
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>'. Tag access is Read Only
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>'. Tag does not meet filtering requirements

Automatic Tag Database Generation Error Messages
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 is not responding

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: Memory allocation error

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 Runtime.
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 is 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 is disabled.
0x013	19	Port is 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 is unreachable.
0x500	1280	ROUTERERR_NOLOCKEDMEMORY No locked memory can be allocated.
0x501	1281	ROUTERERR_RESIZEMEMORY The size of the router memory could not be changed.
0x502	1282	ROUTERERR_MAILBOXFULL The Mailbox of the AMS Port has reached the maximum number of possible entries. The current sent message was abolished.
0x503	1283	ROUTERERR_DEBUGBOXFULL The mailbox of the AMS debugger port has reached the maximum number of possible entries. The sent message will not be displayed in the debug monitor.
0x504	1284	ROUTERERR_UNKNOWNPORTTYPE
0x505	1285	ROUTERERR_NOTINITIALIZED Router is not yet initialized.
0x506	1286	ROUTERERR_PORTALREADYINUSE The desired port number is already assigned.
0x507	1287	ROUTERERR_NOTREGISTERED The AMS Port is not registered.
0x508	1288	ROUTERERR_NOMOREQUEUES The maximum number of AMS Ports has been reached.

0x509	1289	ROUTERERR_INVALIDPORT
0x50A	1290	ROUTERERR_NOTACTIVATED The TwinCAT Router is not yet active.
0x700	1792	Error class <device error>.
0x701	1793	Service is not supported by the server.
0x702	1794	Invalid index group.
0x703	1795	Invalid index offset.
0x704	1796	Reading/writing is not permitted.
0x705	1797	Parameter size is 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 and etc.).
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 is invalid.
0x712	1810	Server is in invalid state.
0x713	1811	AdsTransMode is 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 is 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.
0x722	1826	The symbol is not active. Release the handle and try again.
0x723	1827	Access is denied.
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.
0x754	1876	An invalid response was received.
0x755	1877	The sync port is locked.
0x274c	10060	A connection attempt failed because the connected party did not properly respond after a period of time, or the established connection failed because the connected host has failed to respond.

0x274d	10061	No connection could be made because the target machine actively refused it.
0x2751	10065	A socket operation was attempted to an unreachable host.

Address Validation

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

Address Validation
Address '<address>' is out of range for the specified device or register
Array size is out of range for address '<address>'
Data type '<type>' is not valid for device address '<address>'
Device Address '<address>' contains a syntax error

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
[Port open failed with Error Code '<#>'](#)
[Unable to import from '<dll>'](#)
[Unable to load '<dll>'](#)

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>'.

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](#)
[Monitoring the global variable '<tag address>' on device '<device name>' in order to update symbol information when a change is detected](#)
[Unable to gather runtime information for device '<device name>'. Reason '<reason>'](#)
[Unable to perform Beckhoff compatibility on device '<device name>' due to memory allocation error](#)
[Unable to synchronize with configuration file '<file name>' on device '<device name>' due to memory allocation error](#)
[Unable to read tag '<tag address>' on device '<device name>'. Requesting symbol information for each transaction](#)

Device '<Device name>' is not responding

Error Type:
Warning

Result:
1. If the tag was being read, then the Read operation will not be performed and the tag will be invalidated.
2. If the tag was being written, then 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.
4. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device setting.

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.
5. Increase the Request Timeout setting so that the entire response can be handled.

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>'. Actual data type is not compatible with tag of type '<data type>'
Unable to read tag '<address>' on device '<device name>'. Address bounds exceeded
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>'. Error Code '<#>'
Unable to read tag '<address>' on device '<device name>'. Memory allocation error
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>'. Tag data size of '<size>' bytes(s) exceeds actual data size of '<size>' byte(s)
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:
1. The cause depends on the error code(s) returned.
2. TwinCAT does not support a read command for error code 1793 and TwinCAT version 2.10 (Build 909) or older.

Solution:
1. The solution depends on the error code(s) returned.
2. For error code 1793 and TwinCAT version 2.10 (Build 909) or older, users should update TwinCAT to the latest version. For instructions on how to do so, contact Beckhoff.

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>'. Actual data type is not compatible with tag of type '<data type>'
Unable to write to tag '<address>' on device '<device name>'. Address bounds exceeded
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>'. Error Code '<#>'
Unable to write to tag '<address>' on device '<device name>'. Memory allocation error
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>'. Tag access is Read Only
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>'. Tag does not meet filtering requirements

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

Error Type:
Warning

Possible Cause:
1. The cause depends on the error code(s) returned.
2. TwinCAT does not support a write command for error code 1793 and TwinCAT version 2.10 (Build 909) or older.

Solution:
1. The solution depends on the error code(s) returned.
2. For error code 1793 and TwinCAT version 2.10 (Build 909) or older, users should update TwinCAT to the latest version. For instructions on how to do so, contact Beckhoff.

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 Twin-CAT 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: Beckhoff TwinCAT DLL(s) necessary for import are not loaded
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: Device returned error code '<#>'
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: 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