

## Documentation TwinCAT ADS Library for LabVIEW 4.4

---

This document describes the usage and installation of the “TwinCAT ADS Library for LabVIEW“ version 4.4.

1	General Information.....	2
2	System Requirements.....	3
2.1	Operating System .....	3
2.2	Software .....	3
2.3	LabVIEW .....	3
3	Installation.....	4
3.1	LabVIEW Library .....	4
3.2	LabVIEW Examples .....	4
3.3	Tools And Documents.....	4
4	TwinCAT System .....	6
4.1	Addressing .....	6
4.2	Data Access.....	6
5	Programming.....	8
5.1	Init Driver.....	8
5.2	Read / Write Session .....	8
5.3	Read / Write Multiple.....	9
5.4	Event Notifications .....	9
5.5	Error Codes.....	11
6	Support .....	12
7	Appendix.....	13

## 1 General Information

The “TwinCAT ADS Library for LabVIEW” allows the exchange of data via the ADS interface from Beckhoff. Data can be exchanged directly with Beckhoff TwinCAT software as well as Ethernet IO systems.

The library provides the following functionality:

- Reading and writing data by symbol names
- Reading and writing lists of symbols
- Reading and writing data by address
- Reading and writing device state
- Reading by event notification
- Browsing symbol table
- Conversion functions for data types

## 2 System Requirements

### 2.1 Operating System

The library can be used on Windows XP, Vista, 7, 8 and 10 in 32 and 64 bit.

### 2.2 Software

The library uses the Beckhoff TcAdsDll.dll for communication. This requires a TwinCAT or Message Router installation on the local system.

To access a TwinCAT system on a remote target, a Message Router installation on the local system is sufficient. Use one of the following free installation packages from Beckhoff:

**TwinCAT 2.x:**

TwinCAT ADS Communication Library

**TwinCAT 3.x:**

TC3 ADS

### 2.3 LabVIEW

The library can be used with LabVIEW 2010 to LabVIEW 2019 in 32 and 64 bit.

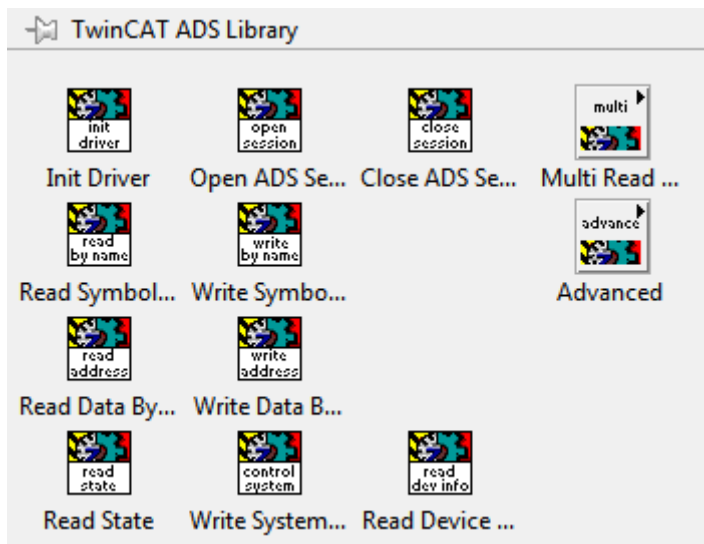
## 3 Installation

### 3.1 LabVIEW Library

The installer installs files directly to the LabVIEW directory (..\program files\National Instruments\LabVIEW xx\..)

The library is installed into the intr.lib folder in the subfolder "\_Ackermann Automation\TwinCAT ADS Library".

The functions palette is installed under Instrument Drivers.



Also a shortcut to Examples and documents is installed in the Start Menu under Programs -> Ackermann Automation -> TwinCAT ADS Library

### 3.2 LabVIEW Examples

The examples are installed in the LabVIEW examples folder  
“..\Program Files\National Instruments\LabVIEW xx\examples\Ackermann Automation\ TwinCAT ADS Library“.

### 3.3 Tools And Documents

All further tools and documents can be found in:

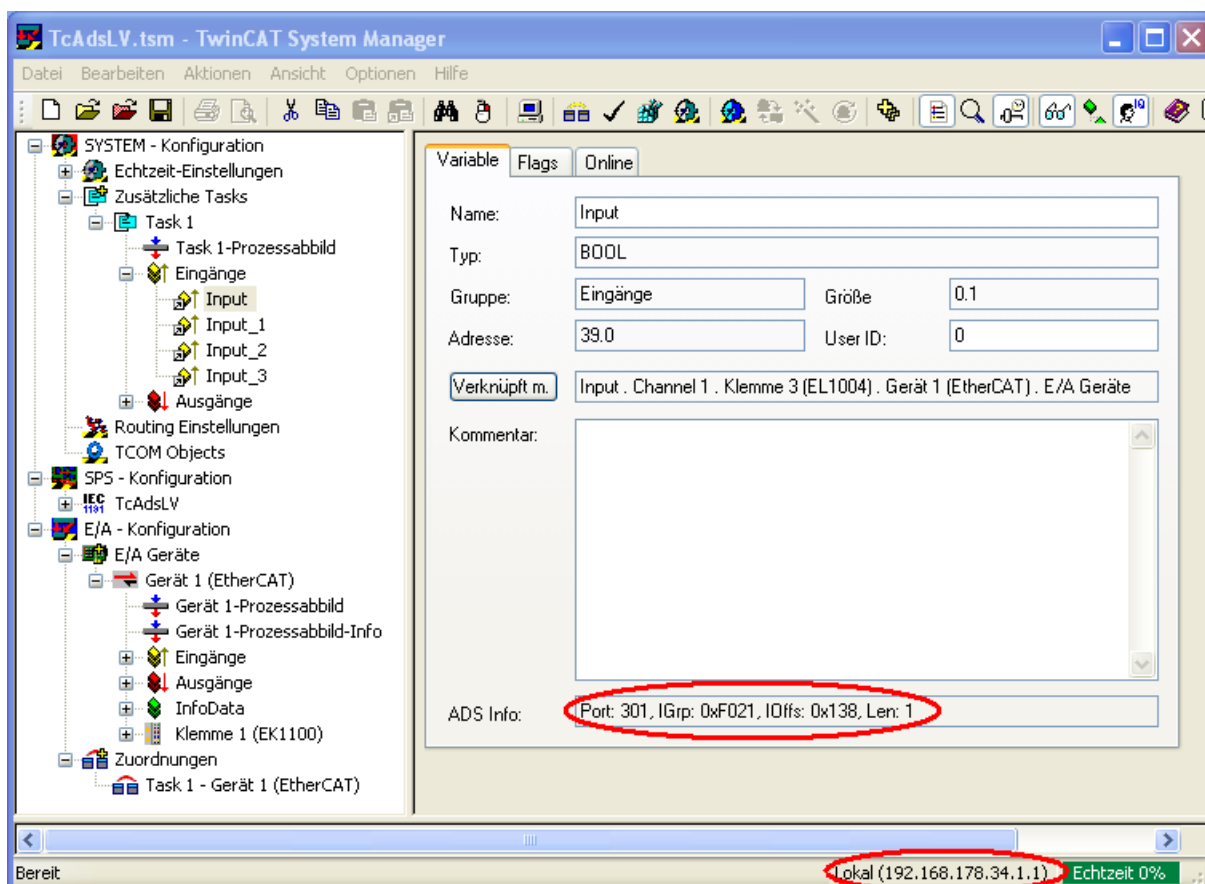
“..\Program Files\Ackermann Automation\ TwinCAT ADS Library “

## 4 TwinCAT System

### 4.1 Addressing

The TwinCAT System provides data access by ADS interfaces to the PLC and the IO system. The ADS interface uses a Net Id and port to identify a device.

The Net Id is an address value consisting of 6 bytes separated by a dot (example: "192.168.1.100.1"). The Net Id is displayed in the bottom line of the TwinCAT System Manager



The port identifies the system like a PLC or an IO task. The PLCs in TC2 have the ports 801, 802, ..., the IO tasks have the ports 301, 302, .. The PLCs in TC3 have the ports 851, 852, ..., the IO tasks have the ports 351, 352, ..

### 4.2 Data Access

The data access can be done by using symbol names or addresses. When symbols are used for IO tasks, symbol creation has to be activated in TwinCAT.

All data is stored in registers that can be accessed by an Index Group and Index Offset. Access by symbol is done by requesting a symbol handle from the ADS server, which corresponds to the registers of the symbol.

It is recommended to use access by symbols, because changes in the register address space are updated in the symbol table. So no change in address configuration has to be done.

**If PLC or task changes are made, that change the data types of the used symbols, make sure to stop the library before the change and restart the library after the change! If the library is running while doing the PLC change, unpredictable results can occur!**

## 5 Programming

This chapter describes the principals of programming with the library. The detailed VI information can be found in the help file.

### 5.1 Init Driver

The function “Init Driver” has to be called before all other function calls.

This function initializes the driver and does a license check.

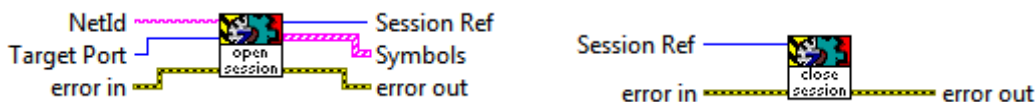
If no valid license number is entered, the library works in demo mode for 10 minutes.



### 5.2 Read / Write Session

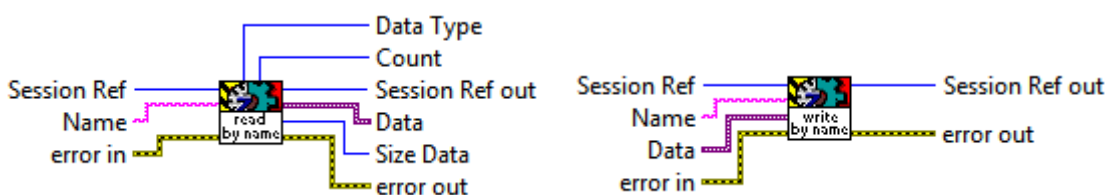
Data access with an ADS device is organized in a ADS session. This means that one ADS port is opened on the local system to communicate with a target port.

On opening the session, the symbol table of the target port is read automatically. This contains the names and address spaces which can be used for the read and write functions.



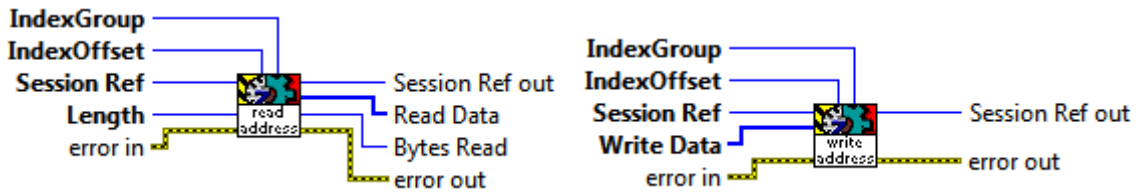
The read and write by symbol name functions use the symbol table information to access data values. Because the data type information is contained in the symbol table, these functions use variant data type and convert it internally to the correct type.

When struct data is used, the data is read and written as byte array data. This has then to be converted to the data types contained in the struct. The conversion functions can be used to convert from / to the correct data types.





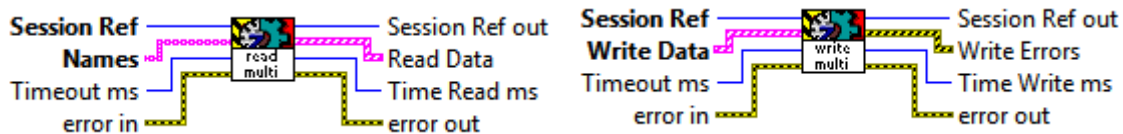
The read and write by address use Index Group, Index Offset and length to access the data. This allows to read whole data areas. The conversion functions can be used to convert from / to the correct data types.



All data access functions are synchronous calls. This means a request is sent and a reply has to be received before the execution continues. This adds up wait times for multiple read and write calls in one session. Use multiple read / write functions for faster data access.

### 5.3 Read / Write Multiple

The read and write multiple functions provide faster data access than the normal read write functions. Multiple symbols or registers are handled in one data access call.

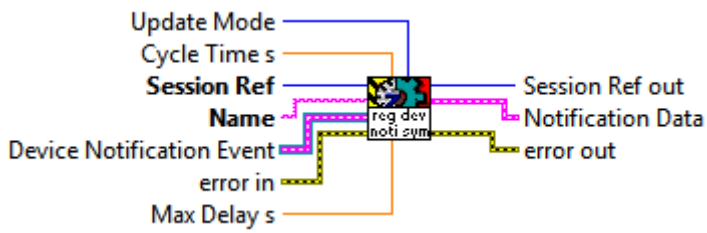
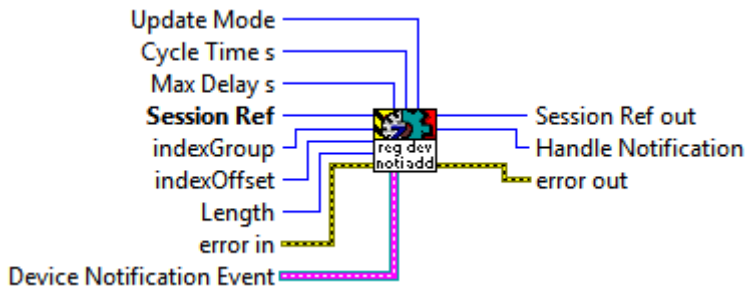


### 5.4 Event Notifications

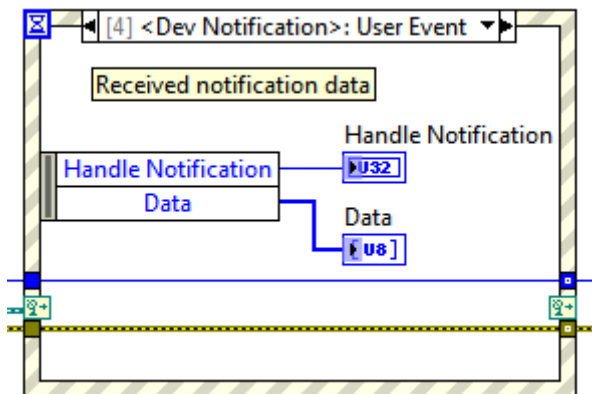
If values from a PLC are to be read continuously, then it is inefficient to use polling read functions, since this function must be called cyclically. By registering notifications (messages), an ADS server can be made to transmit values via ADS to an ADS client. It can be selected if the ADS server is to transmit the values cyclically, or only on value changes.

Since the number of notifications is limited, you should ensure the notifications no longer required by your program are unregistered.

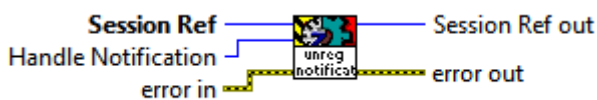
The notification registration can be done by symbol or by address.



The notification data is received in a event structure.



Notifications have to be unregistered after use. Otherwise the ADS server will keep sending data.



## 5.5 Error Codes

The following error codes are used in the library.

5005: Init Driver not called  
5008: Could not open port  
5010: Demo time exceeded  
5012: Session ref not valid  
5020: Error parsing multi read  
5021: Error parsing multi write  
5024: No symbols found  
5025: Symbol not found  
5026: Data size mismatch symbol 'x'  
5055: Pointer not valid

Additionally the Beckhoff ADS error codes are used.

## 6 Support

For support contact:

**Ackermann Automation GmbH**

Kelsterbacher Strasse 15-19

60528 Frankfurt am Main

[www.ackermann-automation.de](http://www.ackermann-automation.de)

Tel.: +49(0)69-40562742

Fax: +49(0)69-40562816

[info@ackermann-automation.de](mailto:info@ackermann-automation.de)

## 7 Appendix

TwinCAT is registered trademark of Beckhoff Automation GmbH.