

# Niagara AX + N4 Beckhoff ADS Driver

**Version V1.2**

## Document information

Document name:	Niagara AX Beckhoff ADS Driver Guide-en new.docx
Directory:	P:\4 Intern\04 Produkte\02 saturnMIS\11 Dokumentation\01 Leitsystem
Status	Freigegeben
Author:	RDS / R

## Changes

Version	Date	Abbr.	Change
V1.0	03.10.2013	fch	Document created
V1.1	29.04.2015	fch	
V1.2	27.11.2018	fch	PLC Comments, array and complex types

© Akt. Jahr (2016) by pi-System GmbH

Alle Rechte vorbehalten. Kein Teil dieses Dokuments darf in irgendeiner Form oder mit irgendwelchen Mitteln, elektronisch, mechanisch, als Fotokopie, Aufzeichnung oder auf andere Weise ohne vorherige schriftliche Genehmigung von pi-System GmbH reproduziert oder übertragen werden.

All rights reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without prior written permission of pi-System GmbH.

## Content

---

Content.....	2
Illustrations .....	3
Tables .....	3
1 Beckhoff ADS Driver Installations.....	4
1.1 License requirements.....	4
2 Installation .....	4
3 Beckhoff ADS Quick Start .....	4
3.1 Configure the Beckhoff Ads Network.....	4
3.2 Add Beckhoff Ads devices.....	5
3.3 Create Beckhoff Ads proxy points.....	6
4 TwinCAT ADS Connectivity.....	6
5 Use the PLC comments .....	7
6 Discover Arrays and complex types.....	8

## Illustrations

---

Figure 1: TPY file property .....	8
Figure 2: Discovered array variables .....	8

## Tables

---

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

# 1 Beckhoff ADS Driver Installations

## 1.1 License requirements

To use the Beckhoff ADS driver, you must have a target Niagara AX host (JACE or Supervisor) that is licensed with the feature `beckhoffAds`.

In addition, note that other limits on devices and proxy points may exist in your license.

The following parts are available:

- |                    |   |
|--------------------|---|
| • PI-DR-ADS-500    | Driver for Beckhoff Ads 500 Points            |
| • PI-DR-ADS-ADD500 | Driver for Beckhoff Ads Additional 500 Points |
| • PI-DR-ADS-DEMO   | Driver for Beckhoff Ads Demo                  |

## 2 Installation

From your PC, use the Niagara Workbench 3.*n.nnn* to install the module `beckhoffAds` in the host.

For details, see “Software Manager” in the *Platform Guide*.

Following this, the remote JACE is now ready for Beckhoff ADS configuration in its running station, as described in the rest of this document.

## 3 Beckhoff ADS Quick Start

### 3.1 Configure the Beckhoff Ads Network

To configure the BacnetNetwork, perform the following main tasks:

#### **Add a Beckhoff Ads Network**

**Note:** As an alternative to the procedure below, you can simply copy (drag and drop) a `BeckhoffAdsNetwork` from the `beckhoffAds` palette in Workbench, placing it in the station's Config, Drivers container.

Use the following procedure to add a `BeckhoffAdsNetwork` under the station's Drivers container:

- |        |  |
|--------|--|
| Step 1 | Double-click the station's Drivers container, to bring up the <b>Driver Manager</b> .  |
| Step 2 | Click the <b>New</b> button to bring up the New Device Network dialog. For more details, see “Driver Manager New and Edit” in the Drivers Guide. |
| Step 3 | Select “BeckhoffAdsNetwork,” number to add: 1, and click <b>OK</b> . This brings up a dialog to name the network.                                |
| Step 4 | Click OK to add the <code>BeckhoffAdsNetwork</code> to the station.  |

You should have a `BeckhoffAdsNetwork` named “BeckhoffAdsNetwork” (or whatever you named it), under your Drivers folder.

#### **Configure the Beckhoff Ads Network**

You must define two essential properties:

- The AMS Net ID of your device
- The AMS port for your device (default 801)

To configure these properties, use the following procedure:

- |        |   |
|--------|---|
| Step 1 | Right-click the <code>BeckhoffAdsNetwork</code> and select <b>Views &gt; Property Sheet</b> . This produces the network's property sheet. |
| Step 2 | In the <b>Source Ams Net Id</b> field, change the entry to a valid AMS Net Id for your device.  |
| Step 3 | You can leave the <b>Source Port</b> field to its default value (801).  |

## 3.2 Add Beckhoff Ads devices

After adding a Beckhoff Ads Network, you can use the network's default device manager view to add the appropriate Beckhoff Ads devices.

### To discover Beckhoff Ads Devices

Perform this task to discover Beckhoff Ads devices.

**Note:** *discovering the devices will not work if the station is running a machine with TwinCAT System Manager, since the TwinCAT System Manager is occupied the UDP port needed for the communication.*

- Step 1 In the Nav tree or in the Driver Manager view, double-click the network, to bring up the device manager.
- Step 2 When the discovery job completes, discovered Beckhoff Ads devices are listed in the top pane of the view, in the Discovered table. The bottom pane is a table of devices that are currently mapped into the Niagara station – initially, this table will be empty.

### To add discovered Beckhoff Ads Devices

Perform this task to add discovered Beckhoff Ads devices to your station database.

- Step 1 You can map a discovered device in the station in a number of ways:
  - Drag it from the Discovered pane to Database pane (brings up an **Add** dialog).
  - Double-click it in the Discovered pane (also brings up an **Add** dialog).
  - Click to highlight in the Discovered and then press the **Add** button.

This works the same as in other driver's Device Manager views.
- Step 2 When the Add dialog appears, you can edit the configuration of the Beckhoff Ads device object before it is added in the Niagara station. Initial property values are determined from the device (and are typically acceptable).
- Step 3 When you have a Beckhoff Ads device component configured properly for your usage, click **OK**.  
The Beckhoff Ads device is added to the station and appears listed in the Database pane.

### To add a Beckhoff Ads device in the network

Use the following procedure to add a Beckhoff Ads device in the network.

- Step 1 In the Nav tree or in the Driver Manager view, double-click the network, to bring up the device manager.
- Step 2 Click the **New** button to bring up the New device dialog.
- Step 3 Select for number to add: 1 (or more, if multiple) and click **OK**.  
This brings up a dialog to name the device(s), enter the IP-address, the Target Ams Net Id and the Ping Target Port.
- Step 4 When you have a Beckhoff Ads device component configured properly for your usage, click **OK**.  
The Beckhoff Ads device is added to the station and appears listed in the Database pane.
- Step 5 Bring up the property sheet for the added device and review the properties.

### 3.3 Create Beckhoff Ads proxy points

As with device objects in other drivers, each client Beckhoff Ads device has a **Points** extension that serves as the container for proxy points. The default view for any Points extension is the Point Manager (and in this case, the "**Beckhoff Ads Point Manager**"). You use it to add Beckhoff Ads proxy points under any Beckhoff Ads device.

For general information, see the "About the Point Manager" section in the *Drivers Guide*.

#### Using online Discover to add Beckhoff Ads proxy points

- Step 1 Bring up the **Beckhoff Ads Point Manager**.
- Step 2 Click the **Discover** button to bring up the Beckhoff ADS Import Options dialog.
- Step 3 Choose a discovering method. You have the following choices:
  - Ads Service
  - Tpy File
- Step 4 Set the other filter properties to your needs. Only the points, which passed the filter, will be listed in the discovered table.  
 The properties "Var Name Filter", "Type Desc Filter" and "Comment Filter" have to be regular expressions pattern.  
 The properties "Tpy File" and "Tpy To Upper Case" is only required if you choose "Tpy File" as Discovering Method.  
**Note:** if you leave the filter with default values, all supported variables will be discovered. Therefore it will be useful to set a judicious filter to limit the discovered variables.
- Step 5 Click **OK** to initiate the discovery process.  
 A progress bar appears at the top of the view, and updates as the discovery occurs.
- Step 6 Select the points you wish to add and click the **Add** button.
- Step 7 When the **Add** dialog appears, you can edit the configuration of the proxy point's **BeckhoffAdsProxyExt** before it is added in the Niagara station. Initial property values are determined from the learned object (and are typically acceptable).  
 Note the following about entries in the **Add** dialog:

#### Manually adding Beckhoff Ads Proxy points

You can manually add Beckhoff Ads proxy points, using the **New** button in the Beckhoff Ads Point Manager, or by dragging from the beckhoff Ads palette.

## 4 TwinCAT ADS Connectivity

For more details, see the Infosys homepage from Beckhoff:

[http://infosys.beckhoff.com/english.php?content=../content/1033/tcconnectivity/html/tcconnectivity\\_intro.htm&id=17122](http://infosys.beckhoff.com/english.php?content=../content/1033/tcconnectivity/html/tcconnectivity_intro.htm&id=17122)

The Beckhoff Ads Driver implements the ADS protocol over TCP/IP native. The Beckhoff ADS connectivity DLLs are not needed, so the driver will work without TwinCAT (i.e. also of a JACE/HAWK).

## 5 Use the PLC comments

Here are examples of comments:

```
(* DisplayName=#Anf. von #Licht
|PATH=#L11\#Zustand|Facets=trueText=%lexicon(pi s#Lexicon#Creator: #Ein)%, falseText=%lexicon
(pi s#Lexicon#Creator: #Aus)%|history=history: #Boolean#Cov#History#Ext@history#Name=%parent.
name%|Access=R *)
```

```
(* DisplayName=#V#L-#Temperatur primär
#W#T|PATH=#H01\#Messwerte|Facets=units=°C, precision=1|history=kit#Pi #System: #Numeric#Cov#I
nterval #History#Ext@history#Name=%parent.name%@change#Tolerance=0.5@interval=180000|Access
=R *)
```

```
(* DisplayName=#V#L-#Temperatur primär #W#T #Grenzwert
oben|PATH=#H01\#Störungen|Facets=trueTextAlarm, falseText=OK|
Alarm=TRUE@source#Name=%parent.parent.display#Name% -
%parent.display#Name%@alarm#Class=#Alarm#Class#Priority|Access=R *)
```

The comment can contain many definitions. The definitions are separated with a special character. This should be | but the characters | and ~ are supported to.

Beckhoff returns all the text on discovering uppercased. So, we need to escape uppercase letter. The escape character is #.

The syntax is `PropertyName=PropertyValue`

You can set the following properties via comment:

- **DISPLAYNAME:** will be set in the proxy extension slot “Long Text For Lexicon” and to the display name.
- **PATH:** you can, with this property, build a folder structure automatically. In the example above, a folder “L11” and a subfolder “Zustand” will be automatically added and the point will be added in the folder “Zustand”.
- **FACETS, DEVFACETS:** you can set the point facets (resp. the device facets of the point) with this property. The Syntax of the value is
  - For Boolean: `trueText=xxx,falseText=yyy`
  - For Numeric: `units=°C,precision=1,min=0,max=100` (precision, min and max are optional)
  - You can also use the same syntax from Niagara to encode Facets
- **ACCESS:** R for read only, W for writable
- **HISTORY:** this is a powerful feature, because the import process will create automatically history extensions for you.  
 The syntax is more complex. It contains several options. There are separated with the character @  
 First is declared the type of the extension. For example: `“history: #Boolean#Cov#History#Ext”`  
 (notice the escape character # for uppercase).  
 Then you can define properties to be set to the extension. For example,  
`change#Tolerance=0.5@interval=180000`
- **ALARM:** this is almost the same like the HISTORY syntax. Instead of extension type (first string in the definition), you can define the alarm condition. For example:
  - For Boolean: `TRUE` or `FALSE` or `==0` or `!=0` or `==1` or `!=1` or `>0`
  - For Numeric: `>lowLimit` or `<highLimit`
  - For Enum: this is not supported yet. You can nonetheless use the feature, the extension will have default settings
 Like for HISTORY, you can define the properties of the extension after the condition (separated with @)

## 6 Discover Arrays and complex types

Unfortunately, the data types cannot be discovered over ADS. For now, the only way is to get the types out from the TPY file. This file is generated automatically from Beckhoff PLC Control as you compile your project. You can follow these steps:

- At first, you will need the actual TPY file. Transfer this file to your Niagara Station
- In the device properties you will find a property named "Tpy File". Set this property to the Ord of the TPY file in the station.

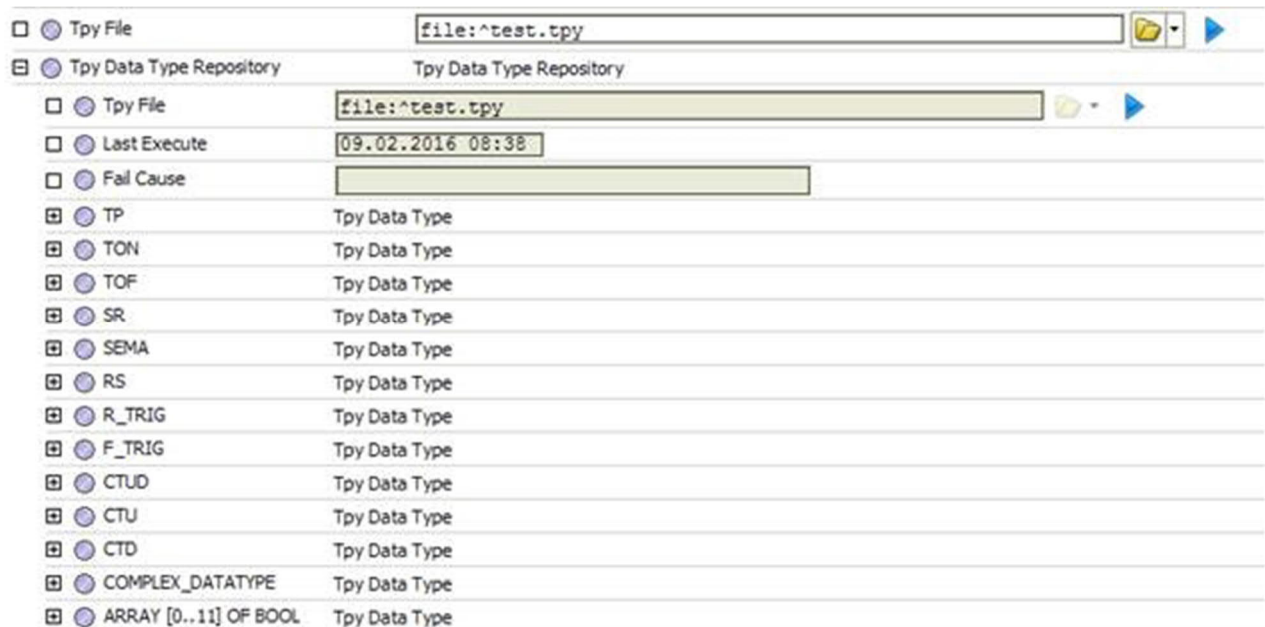


Figure 1: TPY file property

- Go to points and click "discover". You can now discover either over ADS or from TPY.
- Set the option "Reload Data Types From Tpy" to TRUE. This will cause the device to rebuild the "Tpy Data Type Repository". Alternatively, you can invoke the action "Reload Tpy File" on the device.
- After discovering you should see the points as a hierarchical structure in the discover panel. Like this

Entdeckt							
Var Name	Index Group	Index Offset	Value Size	Type Desc	Ads Data Type	Symbol Flags	C
MAIN.COMPLEXVAR	16448	0	16	COMPLEX_DATATYPE	Adst Bigtype	0	
MAIN.COMPLEXVAR.VALUE	16448	0	4	REAL	Adst Real32	0	
MAIN.COMPLEXVAR.ARBOOL	16448	4	12	ARRAY [0..11] OF BOOL	Adst Bigtype	0	
MAIN.COMPLEXVAR.ARBOOL[0]	16448	4	1	BOOL	Adst Bit	0	
MAIN.COMPLEXVAR.ARBOOL[1]	16448	5	1	BOOL	Adst Bit	0	
MAIN.COMPLEXVAR.ARBOOL[2]	16448	6	1	BOOL	Adst Bit	0	
MAIN.COMPLEXVAR.ARBOOL[3]	16448	7	1	BOOL	Adst Bit	0	
MAIN.COMPLEXVAR.ARBOOL[4]	16448	8	1	BOOL	Adst Bit	0	
MAIN.COMPLEXVAR.ARBOOL[5]	16448	9	1	BOOL	Adst Bit	0	
MAIN.COMPLEXVAR.ARBOOL[6]	16448	10	1	BOOL	Adst Bit	0	
MAIN.COMPLEXVAR.ARBOOL[7]	16448	11	1	BOOL	Adst Bit	0	
MAIN.COMPLEXVAR.ARBOOL[8]	16448	12	1	BOOL	Adst Bit	0	
MAIN.COMPLEXVAR.ARBOOL[9]	16448	13	1	BOOL	Adst Bit	0	
MAIN.COMPLEXVAR.ARBOOL[10]	16448	14	1	BOOL	Adst Bit	0	
MAIN.COMPLEXVAR.ARBOOL[11]	16448	15	1	BOOL	Adst Bit	0	

Figure 2: Discovered array variables

- Add the variable you want, like another variable.

Note that the "Tpy Data Type Repository" in the device is only needed to discover the array and complex types. If your Beckhoff project is big, with many complex types and array, the repository could become very big. That's why you can clear it after you finish to add points.



For that, you can invoke the action “Clear Data Types Repo” on the device.  
Another way is to set the option “Clear Data Types Repo After Discover” to TRUE in the discovering options. In this case, the repository will be cleared immediately after discovering the points.