

APP-NOTE 500031

Beckhoff Application Note
Date: 11/27/2001
Document Status: v 1.0

BECKHOFF AUTOMATION LLC

12204 Nicollet Avenue S.
Minneapolis, MN 55337
Phone 952 890 0000
Fax 952 890 2888

Using National Instruments LabView with TwinCAT I/O

OVERVIEW

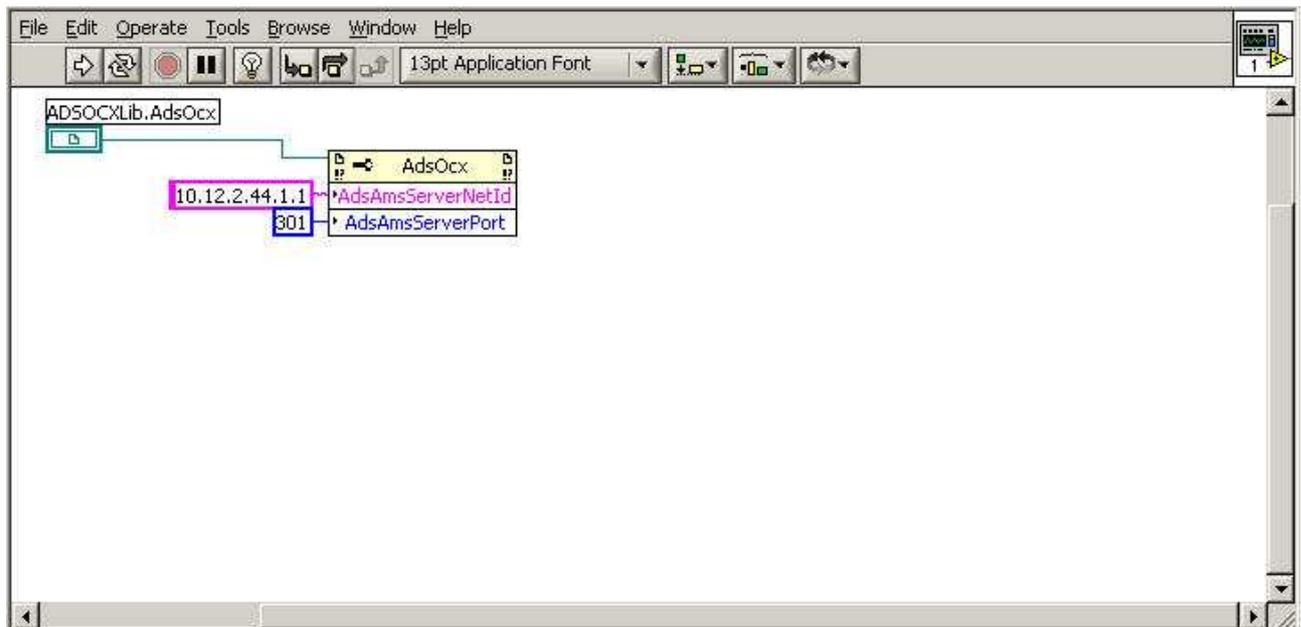
This application note describes how to interface National Instruments LabView with TwinCAT I/O using the ADS.OCX.

SAMPLE FILES USED WITH THIS DOCUMENT

Sample Files	How to Use the Sample Files
example.vi example.wsm	Please refer to LabView documentation for opening the attached example file.

PROCEDURE

- Start LabView and choose to **create a new VI**.
- Right-click in the panel window and **add an Active X container**.
- Right-click within the container that you just added and choose to **insert an ActiveX object**.
- Select **AdsOcx Control** and click 'OK'.
- Right-click in the diagram window and **add an ActiveX Property Node**.
 - o An ActiveX Property Node can be selected at: *Communication / ActiveX / Property Node*.
- Right-click on the Property Node that you just added & select ActiveX Class as **ADSOCXLIB.AdsOcx**.
- Right-click on the Property Node & choose '**Change All to Write**'.
- Right-click over the 'Property' box on the Property Node & choose **Properties / AdsAmsServerNetID**.
- Right-click over the 'AdsAmsServerNetID' box on the Property Node & choose '**Add Element**'.
- Right-click over the new element that you just added & choose **Properties / AdsAmsServerPort**.
- Right-click over each element & choose **Create / Constant**.
- In the constant box that you added for the 'AdsAmsServerNetId', **type in your AMS Net ID number**.
 - o This number can be found in the TwinCAT properties, under the AMS Router tab.
- In the constant box that you added for the 'AdsAmsServerPort', **type in the number '301'**.
- **Wire the ADSOCXLib.AdsOcx box to the input on your AdsOcx Property Node**. At this point, your diagram window should look like the following:



- Now right-click within the diagram window and **add an ActiveX 'Invoke Node'**.
 - o An Invoke Node can be selected at: *Communication / ActiveX / Invoke Node*.
- Right-click on the Invoke Node that you just added & select ActiveX Class as **ADSOCLIB.AdsOcx**.
- Right-click on the Invoke Node. Choose **Methods / AdsReadIntegerConnect**.
- Right-click over each element & choose **Create / Constant**
- Change the input constant for '**nIndexGroup**' to that of the variable that you are reading from TwinCat. You will have to change the input constant in LabView to hex format.
 - o To find this value, click on the variable that you added to your real time task in TwinCat System Manager. Go to the 'Variable' tab on the right. Near the bottom of that window is a line labelled 'ADS Info'. You should see a value labelled **IGrp: 0xF021**. (The last 4 digits may be different.)
 - The last 4 digits of **IGrp: 0x_____** is your index group location in hex.
- Change the input constant for '**nIndexOffset**' to that of the variable that you are reading from TwinCat. You will have to change the input constant in LabView to hex format.
 - o This value is located to the right of the index group number within TwinCAT System Manager.
 - The last digit of **IOffs: 0x_** is your index offset value in hex.
- Change the input constant for '**cbLength**' to that of the variable that you are reading from TwinCat. This will just be a decimal value.
 - o This value is located to the right of the index offset value within TwinCAT System Manager.
 - The digit in **Len: _** is the length of your variable.

- Change the input constant for '**nRefreshType**' to 1, 3, or 4. Choose the type based on the information below. (Taken from the TwinCAT information system.)

TwinCAT ADS-OCX: Reference

ADSOCTXTRANSMODE

ADSTRANS_CLIENTCYCLE	= 1
ADSTRANS_SERVERCYCLE	= 3
ADSTRANS_SERVERONCHA	= 4

Description

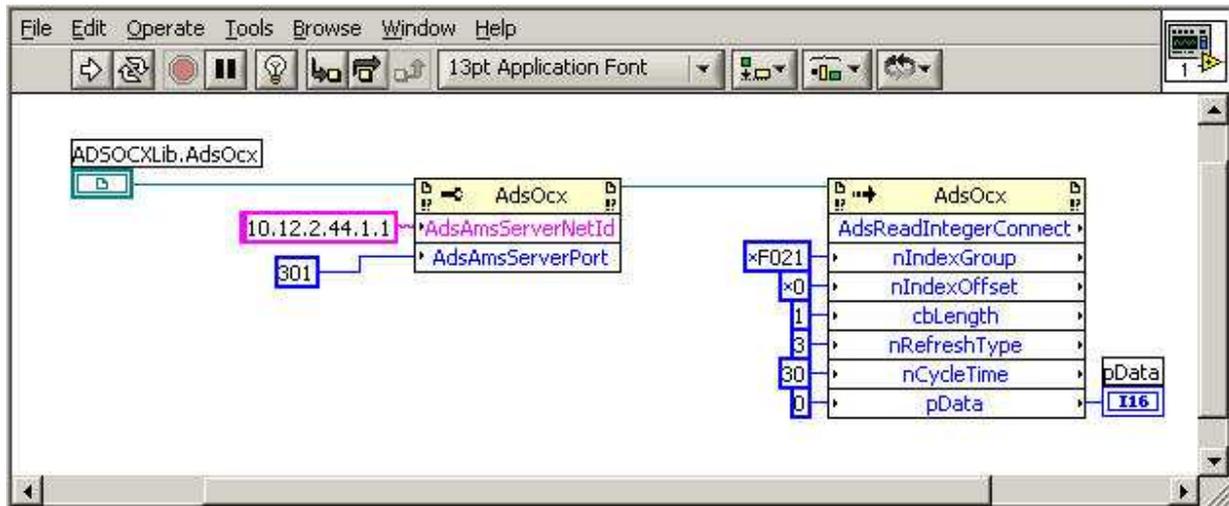
Parameter	Description
<i>ADSTRANS_CLIENTCYCLE</i>	The ADS-OCX executes a write / read command cyclically. The AdsReadConnectUpdateEx() event function is then called. The cycle time is rounded up to a multiple of 55. The shortest time is 55 ms. The timer that initiates the read / write runs in Windows NT/2000 user mode, which means that the time behaviour strongly depends on the loading of the system.
<i>ADSTRANS_SERVERCYCLE</i> (only when reading)	The ADS that has been addressed writes the data cyclically to the ADS-OCX. The AdsReadConnectUpdateEx() event function is then called. The smallest possible time is the cycle time of the ADS device; for the PLC, this is the task cycle time. The cycle time can be handled in 1 ms steps. If you enter 0 ms as the cycle time, then the data is sent to the ADS-OCX with every cycle of the ADS device task.
<i>ADSTRANS_SERVERONCHA</i> (only when reading)	The ADS device that has been addressed then only writes the data to the ADS-OCX if they have changed. The ADS device is sampled at the rate given by the cycle time. The AdsReadConnectUpdateEx() event function is then called. The cycle time can be handled in 1 ms steps. If you enter a cycle time of 0 ms, every change in the variables will be sent to the ADS-OCX. A longer cycle time can be used to reduce the number of data transmissions to the ADS-OCX.

Warning: Too many write / read operations can load the system so heavily that the user interface becomes much slower.

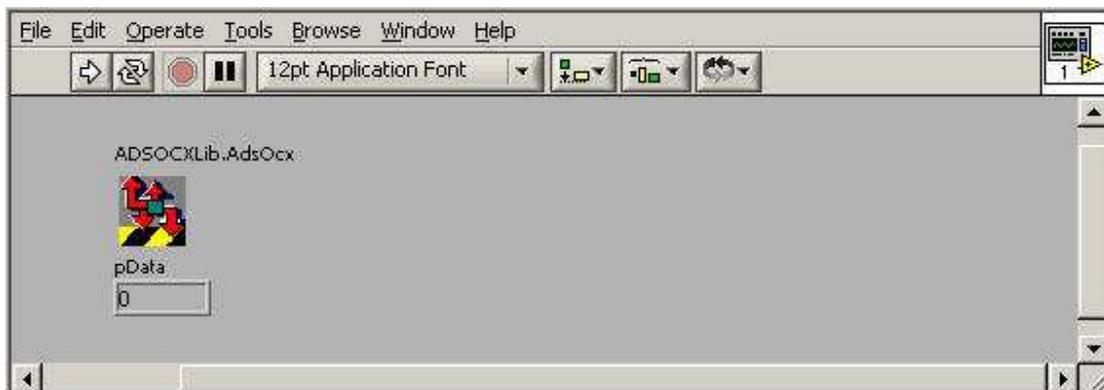
Tip: Set the cycle time to the most appropriate values, and always close connections when they are no longer required.

- Change the input constant for '**nCycleTime**' to a value that makes sense for your system (read the information above). This is also just a decimal value.
- Leave the input constant for '**pData**' as zero. This just creates a variable location to be used to write our data to.

- Right-click on the pData element and choose **Create / Indicator**. This creates an output box in the LabView panel window that will show us the current state of our TwinCAT variable.
- **Wire your labview property node to your AdsReadIntegerConnect node.** At this point, your diagram window should look something like the following:

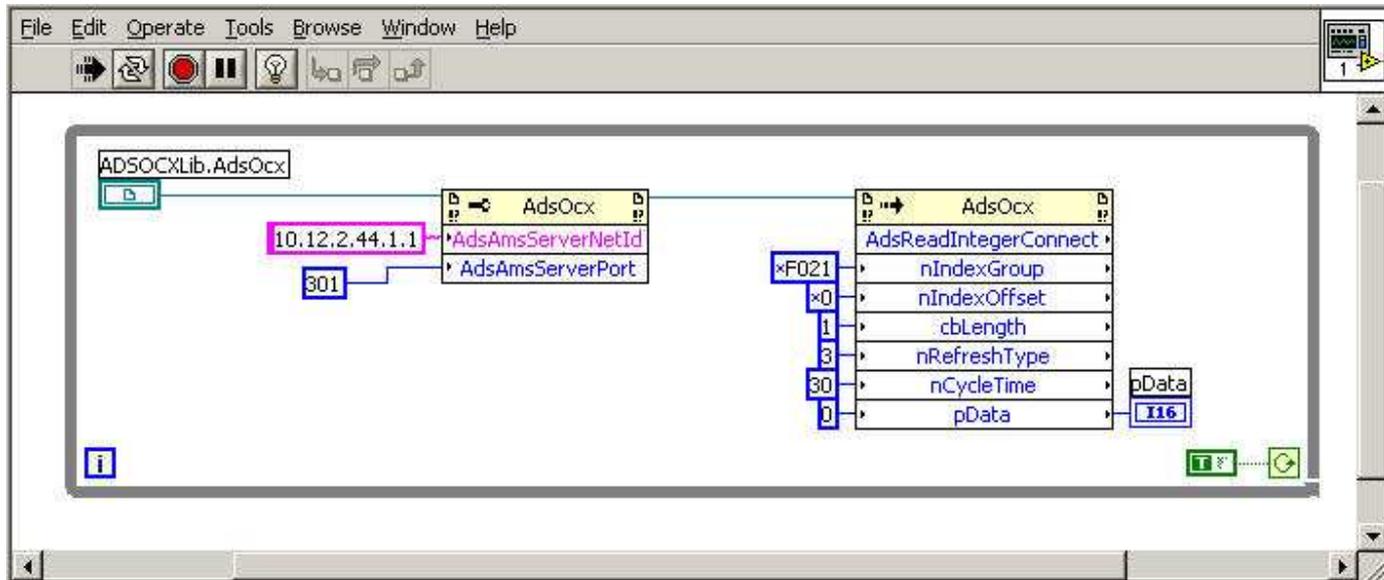


- Your panel window should look something like the following:



- The value shown in the **pData** box is the current value of your TwinCAT variable. In this example, our variable is just a bit, so the value will either be 0 or 1.
- Make sure that the TwinCat System is running (icon green). In TwinCat system manager, look under the online tab of the variable that you referenced and **force the variable to a value of 1**.
- Run the LabView program. The value in the **pData** box should change to 1.
- In order to make the LabView program run continuously so that it keeps updating the variable data, put a while loop around the code in the diagram window and set the

condition to be a true constant. Your diagram window should then look like the following:



- Now run the LabView program again. It should continue to run unless you push the stop icon on the LabView toolbar. Go back into TwinCat System Manager and force a change in the state of your variable a few times. You should be able to see the changes in the **pData** box on your LabView panel.

SUPPORTING DOCUMENTS AND REFERENCES