Documentation EtherCAT Library for LabVIEW 2.9

This document describes the usage and installation of the “EtherCAT Library for LabVIEW“ version 2.9.

1 General Information

The “EtherCAT Library for LabVIEW” allows direct data exchange with EtherCAT slaves. LabVIEW acts as an EtherCAT Master and controls the slaves in the EtherCAT topology. A Connection can be made with a normal network adapter (NIC). The EtherCAT Master can be run with a realtime driver for controlling the NIC. This allows a deterministic bus rate of up to 10 kHz and the usage of Distributed Clocks. As an alternative, the EtherCAT Master can also be run in Windows mode without realtime behavior.

The library provides a wide set of EtherCAT functionality like:

- Cyclic process data exchange
- CAN over EtherCAT (CoE)
- Servo Profile over EtherCAT (SoE) protocol
- Access to Slave EEPROM and Registers
- File Transfer over EtherCAT (FoE) mailbox protocol
- Synchronization with Distributed Clocks (DC) including Master Synchronization (DCM)

2 System Requirements

2.1 Operating system

The library can be used on Windows XP, 7, 8.1 and 10 with 32 and 64 bit.

2.2 PC System

In Realtime Mode
The realtime driver requires PC Systems with ACPI (Advanced Configuration and Power Interface) or APIC (Advanced Programmable Interrupt Controller).

In Windows Mode
No restrictions regarding PC hardware.

2.3 Network adapter

In Realtime Mode
EtherCAT protocol is run on a normal network adapter, which is controlled by a realtime environment. This requires the installation of a realtime driver for the NIC and limits the supported network adapters to chipsets from **Intel** and **Realtek**.

In Windows Mode
No restrictions regarding Network adapter.

2.4 LabVIEW

The library can be used with LabVIEW 2009 to LabVIEW 2016 in 32 bit.

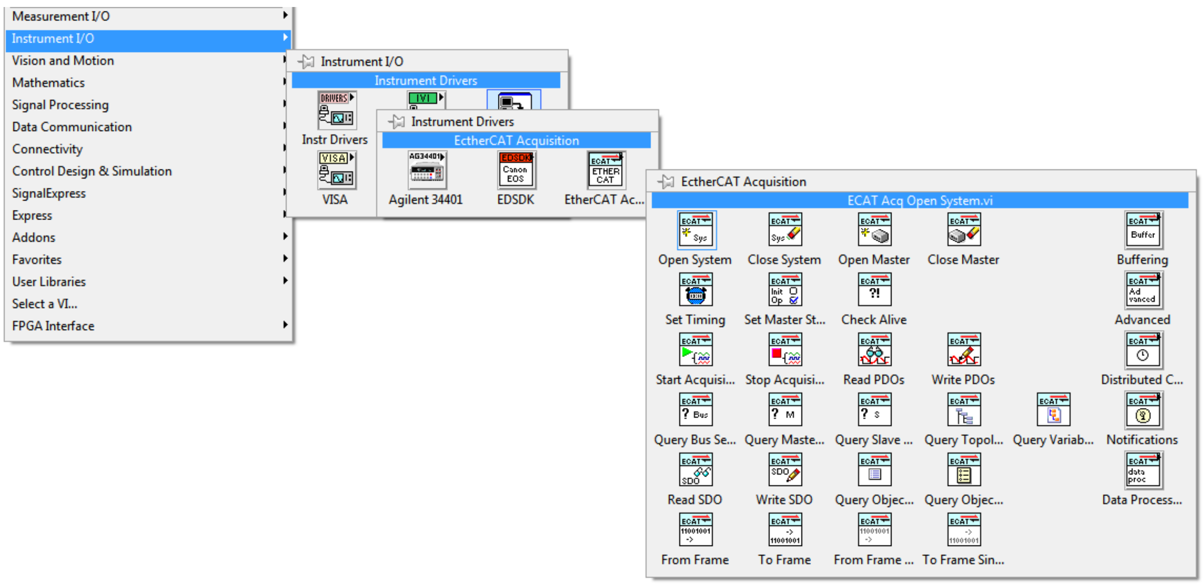
3 Installation

3.1 LabVIEW Library

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

The library is installed into the intr.lib folder in the subfolder "_Ackermann Automation\EtherCAT Acquisition".

The functions palette is installed under Instrument Drivers.



3.2 LabVIEW Examples

The examples are installed in the LabVIEW examples folder “..\Program Files\National Instruments\LabVIEW xx\examples\Ackermann Automation\EtherCAT Acquisition “.

3.3 Tools And Documents

All further tools and documents can be found in:

“..\Program Files\Ackermann Automation\EtherCAT Acquisition Library\2.x“

3.4 Network Card Driver

In Realtime Mode
This topic is covered in the document "Installation Realtime Driver".

In Windows Mode
In Windows mode the WinPcap driver has to be installed. This driver can be downloaded from: www.winpcap.org

4 Basic Programming

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

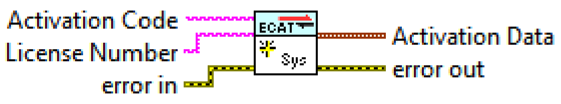
4.1 System Driver

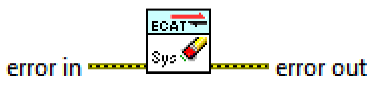
The function “Open System Driver” has to be called before all other function calls.

This function prepares the realtime environment and checks the license activation status. If the license is not using a USB dongle, the license key has to be entered. If no dongle or valid license key is found, the library is running in demo mode for 10 minutes.

The driver has to be closed when finishing execution. This will shut down the resources of the realtime environment.

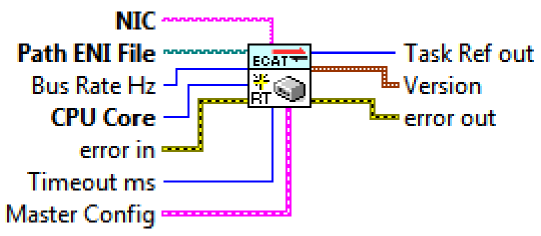
Avoid stopping execution without using the Close System function call!

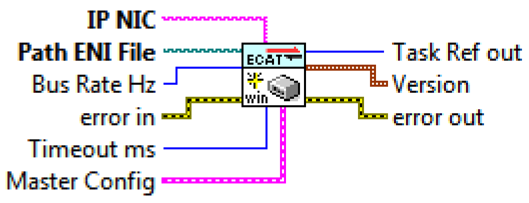




4.2 EtherCAT Master

After the system driver has been opened, the EtherCAT Master can be started. The EtherCAT Master can be run with the Realtime Driver or in Windows mode. Which option is selected is determined by the Open Master function. The Windows mode does not support the whole functionality of the library due to the lack of deterministic timing behavior.





In both cases an ENI file is required to run the master with cyclic process data. This XML file contains the information about the whole EtherCAT system including the slave and process data description. The ENI file is described in the chapter "ENI Files".
All actions that do not require Master state higher PreOp (CoE, memory access, bus scan, etc.) can be done without an ENI file.

If no ENI file path is entered, a basic configuration is created by the master. In this configuration the physical slave addresses start from 1. In common ENI configuration tools like TwinCAT or EC Engineer physical slave addresses start from 1001.

In Realtime Mode
Opening the EtherCAT Master requires the network adapter information and the CPU core selection.

The network adapter (NIC) has to be selected. The network adapter has to be assigned to the realtime (see document "Installation RealTime Driver"). There are 4 different types of network adapter classes supported. Intel 100 and 1000 Mbit NICs and Realtek 100 and 1000 Mbit NICs. The type is identified by the following strings:

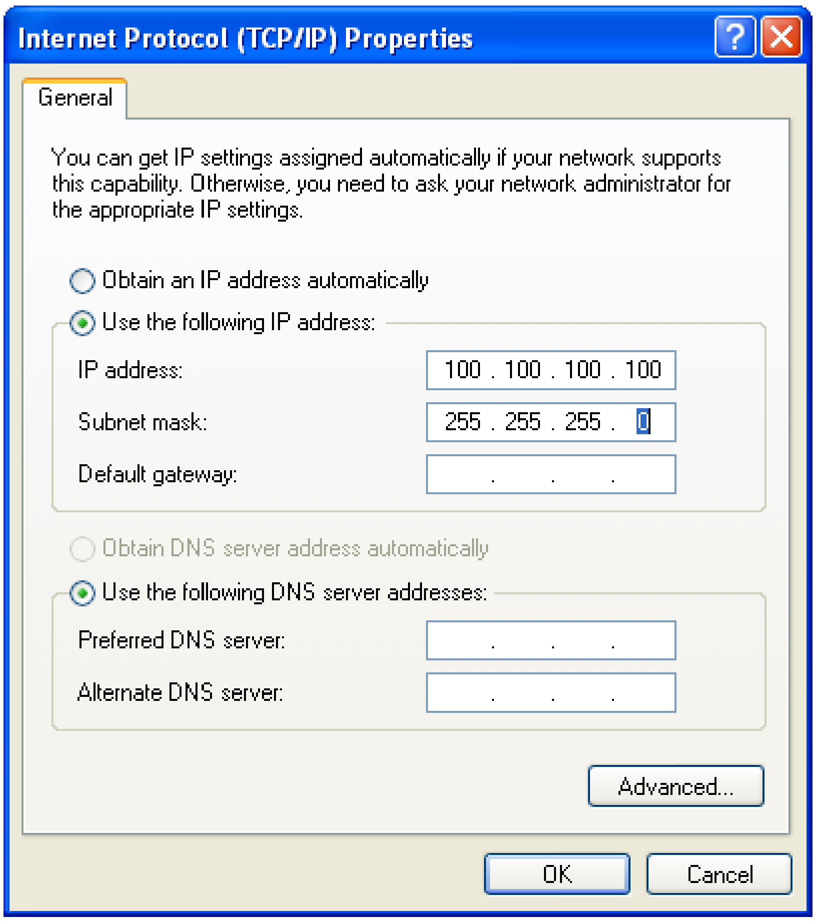
"Intel1000"
"Intel100"
"Realtek1000"
"Realtek100"

The processor parameter defines the CPU core usage of the realtime driver. The realtime driver can be used in two different modes. In shared mode, Windows and the realtime driver share a CPU core. In exclusive mode, the realtime driver uses one CPU core exclusively. This core has to be taken from Windows (see document " Installation RealTime Driver ").

<u>On 64bit Systems, only exclusive mode is available!</u>

In Windows Mode
Opening the EtherCAT Master only requires the network adapter information.

The windows network adapter to be used has to be given a static IP address in the Windows network configuration. This IP address is used to identify the network adapter.


Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

( ) Obtain an IP address automatically
[✓ Use the following IP address:]

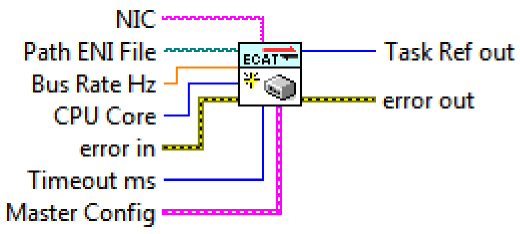
IP address:	100 . 100 . 100 . 100
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	. . .

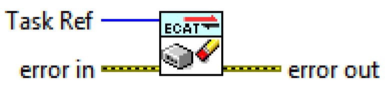
( ) Obtain DNS server address automatically
[✓ Use the following DNS server addresses:]

Preferred DNS server:	. . .
Alternate DNS server:	. . .

The bus cycle rate determines the rate in which the cyclic process data is sent. In Windows mode this rate is limited to 1 kHz with non deterministic Windows timing. In realtime mode the bus cycle rate can be up to 10 kHz.

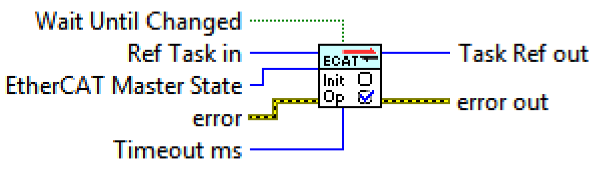
The function returns a task ref, which all other functions use as input.



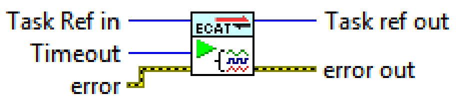


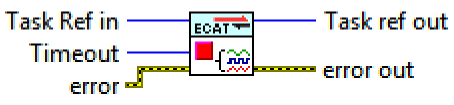
4.3 Bus State

The EtherCAT bus states "Init", "PreOp", "Boot", "SafeOp" and "Op" can be set with the function Set Bus State. The state is set for the master which tries to set all connected slaves into this state. Cyclic process data exchange is executed in the states "SafeOp" and "Op". In "SafeOp" no outputs are set in the slaves.



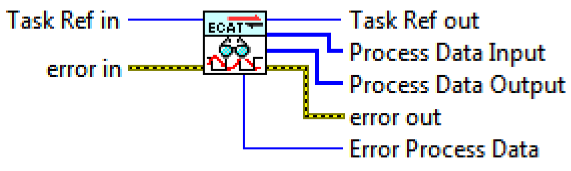
The functions start and stop are shortcuts to "Op" and "Init" state.

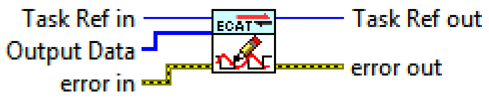




4.4 Reading And Writing Cyclic Process Data (PDO)

The cyclic process data is organized in an input data image and a output data image. These data sections contain all the input and output process data from the slaves in the topology. The single frame read and write calls give access to these data sections.
These functions give direct access and have no buffering. That means, if the bus cycle rate is faster than the LabVIEW loop, process data is lost.



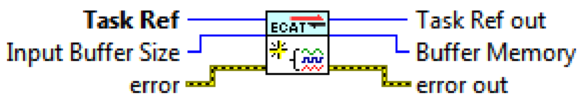


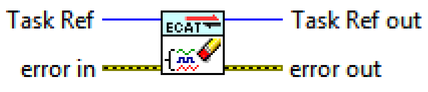
4.5 Buffered Reading Cyclic Process Data (PDO)

Only supported in realtime mode.

With the buffered data exchange of cyclic process data, the cyclic bus rate can be greater than the LabVIEW loop to handle the process data. Data is transferred in packets. Input data can be acquired without data loss.

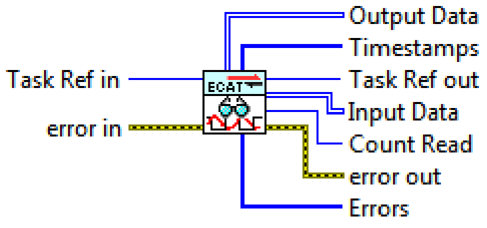
If cyclic process data is to be transferred in buffered mode, a buffer has to be created before. The buffer size is determined in count of frames. So for example a size of 1000 can hold data for 1000 cycles. When running a cycle rate of 1 kHz, the buffer can hold data for 1 second. The input buffer always uses the full input process data.





The read function usees a 2D arrays for the data exchange. A row of the array is the data of one cycle.

The input data contains the data of inputs and outputs of one process data cycle. A timestamp in µs of the time the telegram was sent is taken in the realtime environment and an error value for every process data cycle is transferred.

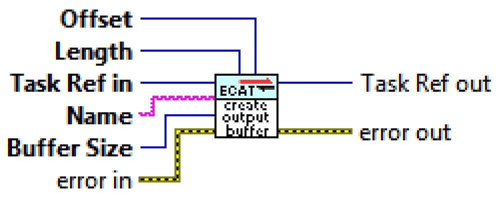


4.6 Buffered Writing Cyclic Process Data (PDO)

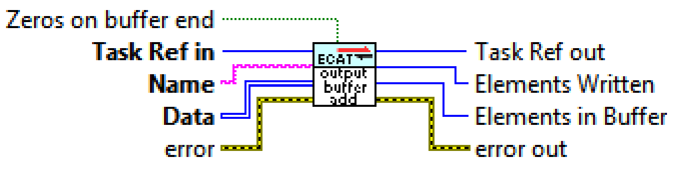
Only supported in realtime mode.

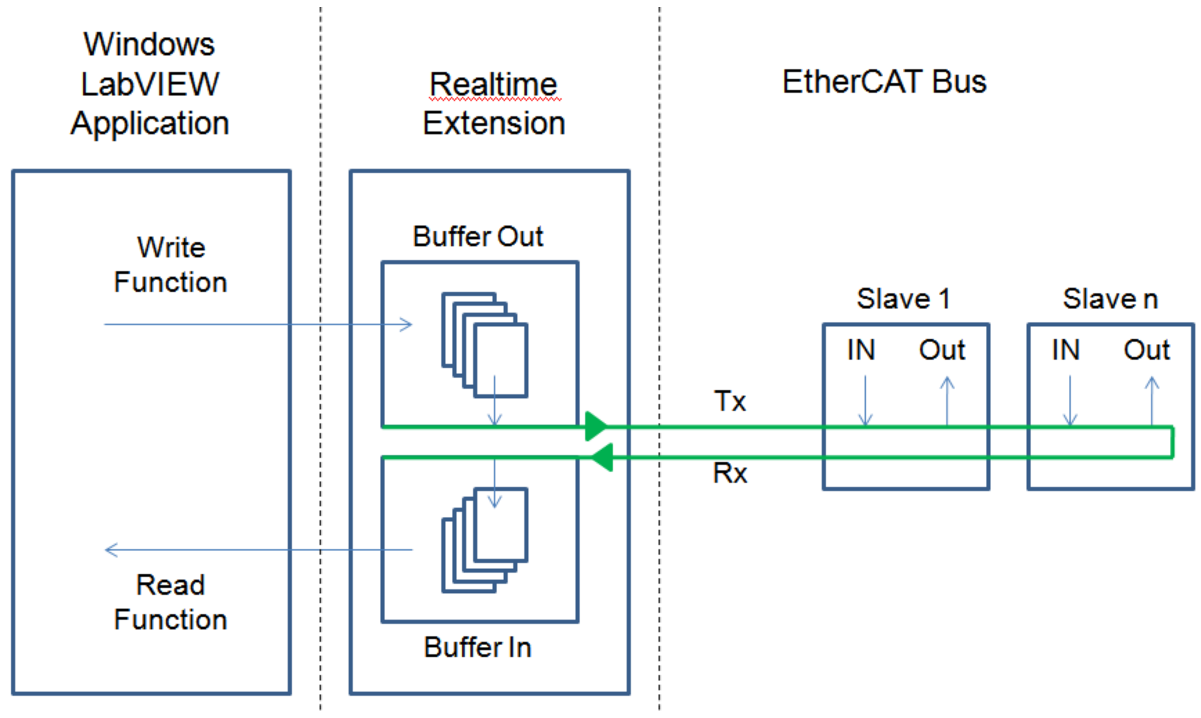
Also output process data can be transferred using buffers. Here it is possible to create multiple buffers for different output process data segments. By this it is possible to have some outputs creating buffered signals, while other outputs respond to fast single write operation.

A buffer is created by giving a name for later reference and the data area.



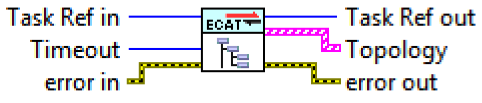
This buffer can be filled with the cyclic process data to write



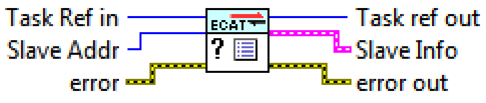


4.7 Topology And Slave Information

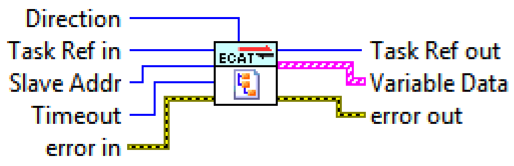
The topology information containing the general slaves data can be queried from the bus. The data is returned as a slave data array.



Specific Slave information can be read online from each slave.

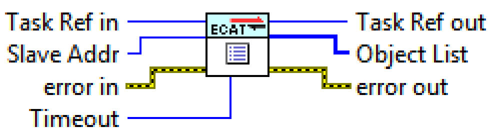


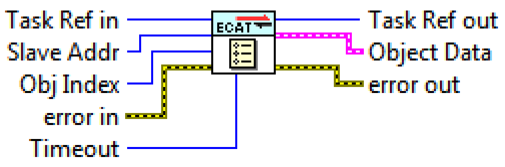
Each slave has a set of input and output variables in the process data. The information about these sets can be queried. The returned data contains the positions in the input and output process data images with byte and bit offsets.



4.8 Object Dictionary

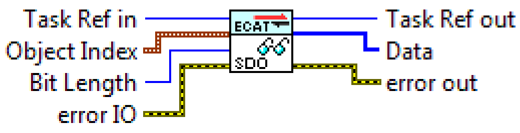
The object dictionaries of connected slaves can be read, if they support mailbox communications.

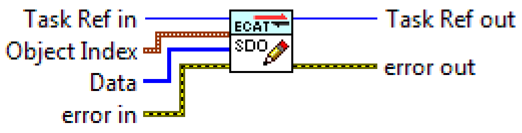




4.9 Reading and Writing Service Data Objects (SDO)

Service data objects (SDOs) can be accessed via CoE. The adressing is done by index and subindex.

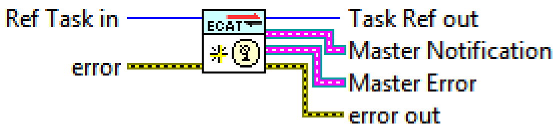


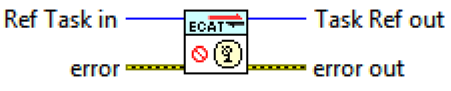


4.10 EtherCAT Master Events

Only supported in realtime mode.

The EtherCAT Master sends notifications and errors as LabVIEW User Events. These Events have to be registered in a LabVIEW event structure to be received.





5 Data Processing Functions

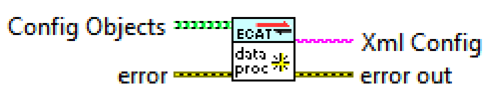
Only supported in realtime mode.

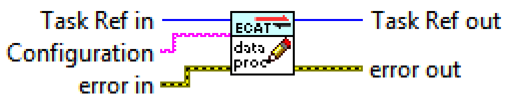
Data Processing Functions are predefined functions that work on the process data and are executed in the realtime environment in the time between receiving and sending of frames. These functions run with the bus cycle rate of up to 10 kHz.

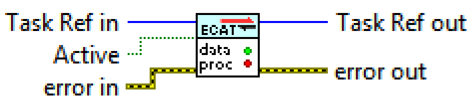
The following functions are available:

- Bit Copy
- Analog Copy
- Limit Monitor
- Boolean Logic

The functions are configured by creating a LabVIEW object array in which different function types can be combined. This object array is converted into a XML string, which is downloaded to the realtime environment. Here the XML configuration is creating the corresponding functions, which can be activated and deactivated.



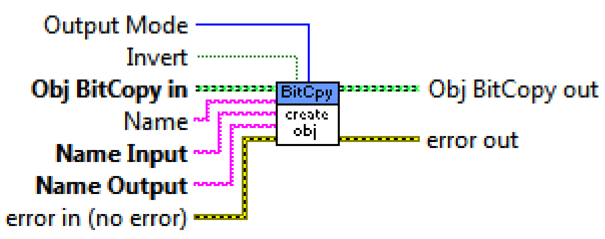




The Xml configuration is generated with the following Vis.

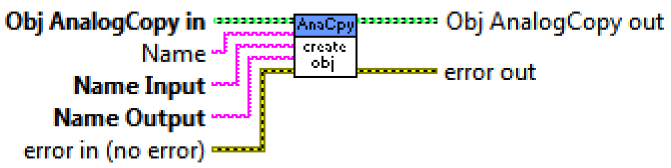
5.1 Bit Copy

The Bit Copy function copies the data of a boolean input variable to a boolean output variable.



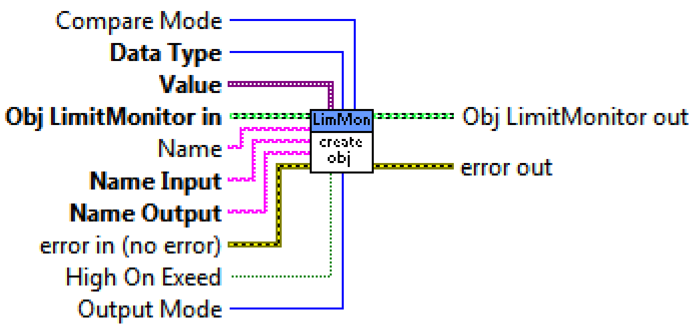
5.2 Analog Copy

The Analog Copy function copies the data of an analog input variable to an analog output variable.



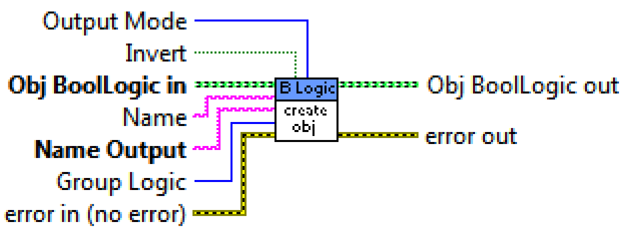
5.3 Limit Monitor

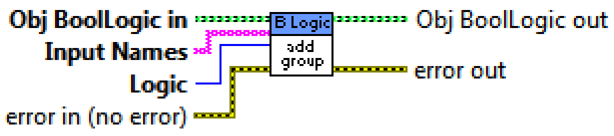
The Limit Monitor function compares a threshold value with a current analog input and writes the result to a boolean output variable.



5.4 Bool Logic

The Bool Logic function evaluates boolean groups and writes the result to a boolean output variable.





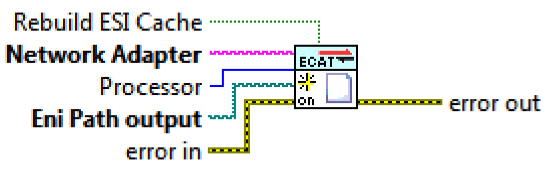
6 ENI Files

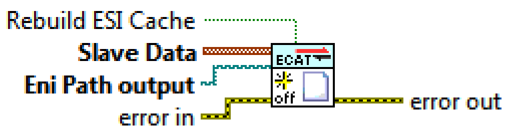
For cyclic process data exchange, the EtherCAT Master needs a valid EtherCAT ENI file (EtherCAT Network Information) that describes the bus topology and the process data setup of all slaves. There are several ways to create these ENI files.

6.1 Creation with LabVIEW

The EtherCAT Acquisition library provides the functionality to create ENI files.

For ENI file creation the slave identification data vendor ID, product Code and Revision number have to be known. There are alternatives for creating the ENI files. One is to scan the topology online and create the ENI file. The other is to enter the slave data offline.





The ENI file creation is based on slave ESI files. Make sure to have the most recent ESI files of your slaves in the ESI file folder. The ESI file folder can be accessed form a shortcut in the Start Menu (Ackernmann Automation -> EtherCAT Acquisition -> 2.x).

Regarding licensing it is distinguished between a basic mode, which comes with the runtime license and a ENI Creation option, which has to be licensed as an addition. The basic mode can only handle straight topologies without EtherCAT junctions. Meaning that all slaves are connected to Port B of the previous slave. This can for example be one Beckhoff EK1100 with connected terminals. Two EK1100 will not work, because the second EK1100 is connected to Port C of the first EK1100.
Also these files only contain the standard (default) process data configuration for the slaves. This standard is defined in the slave ESI files. And Distributed Clocks settings cannot be configured.

See examples "Example - Create ENI File Online Basic.vi" and "Example - Create ENI File Offline Basic.vi" in "..\examples\EtherCAT Acquisition\ EtherCAT Library"

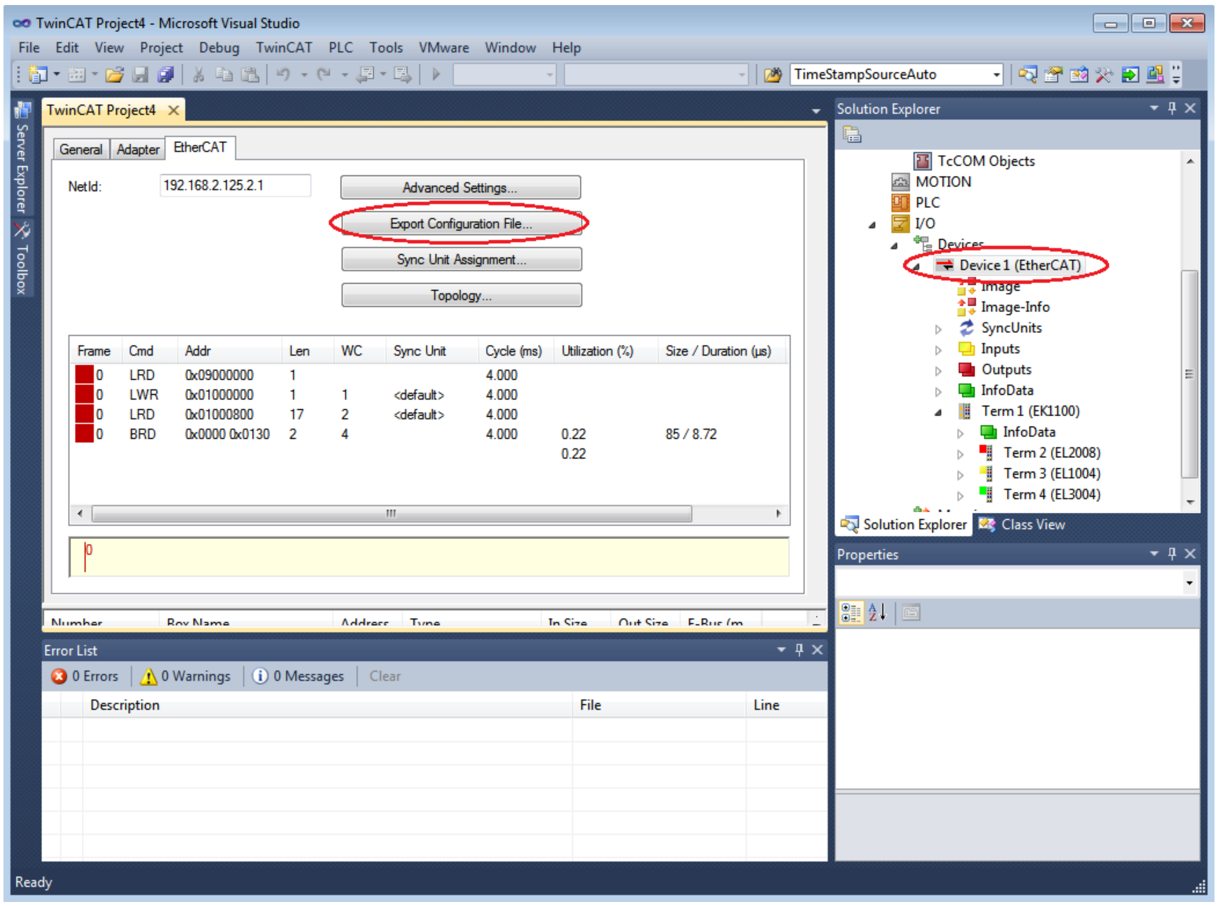
More complex systems can be handled with the ENI Creator Library.

See the examples in "..\examples\EtherCAT Acquisition\ ENI Creation Library"

6.2 Creation with TwinCAT

ENI files can be created in TwinCAT System Manager. The EtherCAT Topology can be entered manually or by bus scan. The export function located in the EtherCAT tab. The exported XML file is the ENI file.

TwinCAT 2.x or 3.x can be used. The TwinCAT System Manager comes with the free TwinCAT CP installation.



NetId: 192.168.2.125.2.1

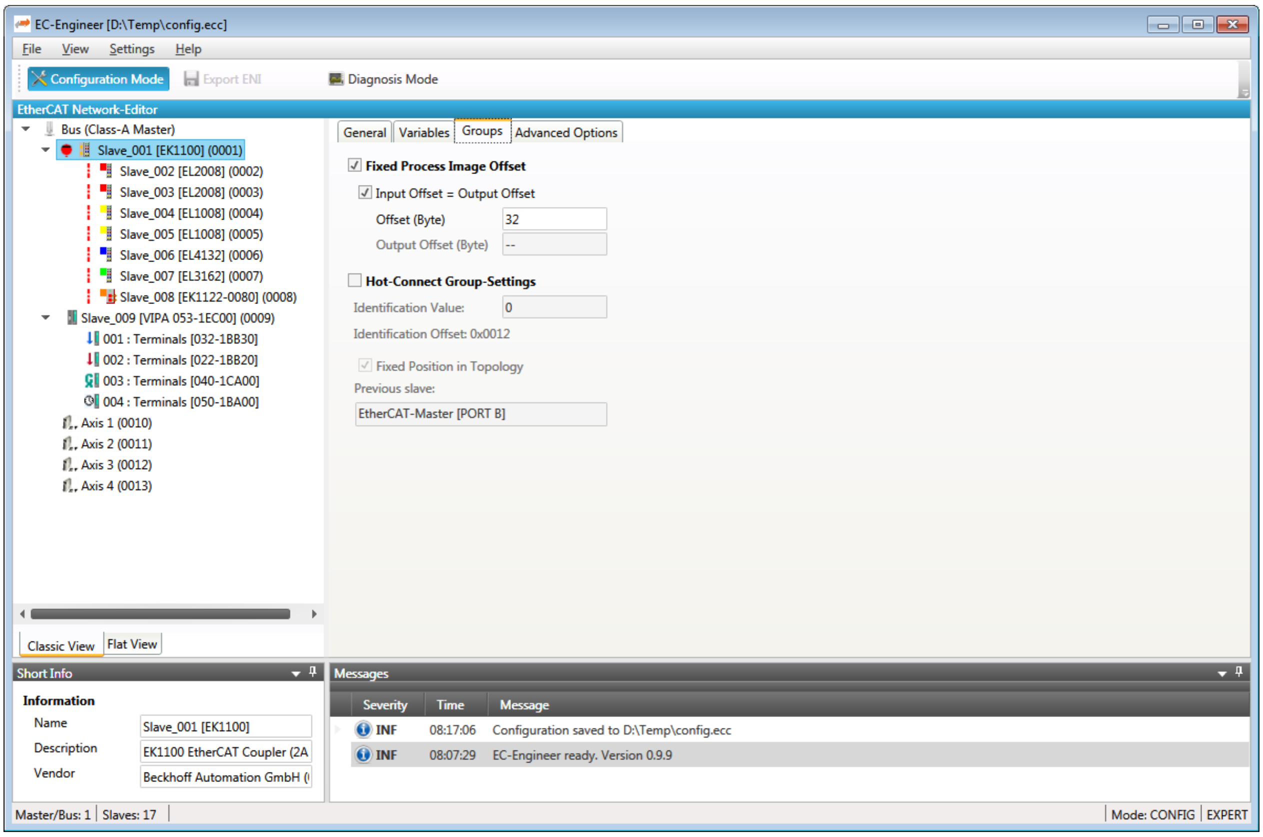
Frame	Cmd	Addr	Len	WC	Sync Unit	Cycle (ms)	Utilization (%)	Size / Duration (µs)
🟥 0	LRD	0x09000000	1			4.000		
🟥 0	LWR	0x01000000	1	1	<default>	4.000		
🟥 0	LRD	0x01000800	17	2	<default>	4.000		
🟥 0	BRD	0x0000 0x0130	2	4		4.000	0.22	85 / 8.72
							0.22	

6.3 Creation with EC Engineer

The EC Engineer software allows the ENI file creation as well as bus diagnosis.

The software works with any network adapter including USB network adapters. Make sure a standard IP address is set for the adapter (No DHCP).

Switch the view to "Expert", to have the full set of functionality.



Bus (Class-A Master)
- Slave_001 [EK1100] (0001)
  - Slave_002 [EL2008] (0002)
  - Slave_003 [EL2008] (0003)
  - Slave_004 [EL1008] (0004)
  - Slave_005 [EL1008] (0005)
  - Slave_006 [EL4132] (0006)
  - Slave_007 [EL3162] (0007)
  - Slave_008 [EK1122-0080] (0008)
- Slave_009 [VIPA 053-1EC00] (0009)
  - 001 : Terminals [032-1BB30]
  - 002 : Terminals [022-1BB20]
  - 003 : Terminals [040-1CA00]
  - 004 : Terminals [050-1BA00]
- Axis 1 (0010)
- Axis 2 (0011)
- Axis 3 (0012)
- Axis 4 (0013)

[✓] **Fixed Process Image Offset**

[✓] Input Offset = Output Offset

Offset (Byte)	32
Output Offset (Byte)	--

[ ] **Hot-Connect Group-Settings**

Identification Value:	0

Identification Offset: 0x0012

[✓] Fixed Position in Topology

Previous slave:

EtherCAT-Master [PORT B]

Information

Name	Slave_001 [EK1100]
Description	EK1100 EtherCAT Coupler (2A
Vendor	Beckhoff Automation GmbH (

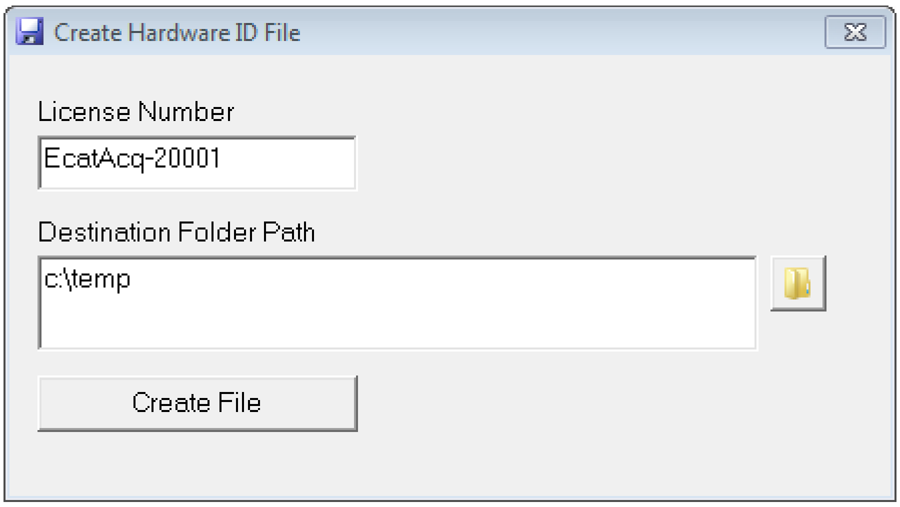
Severity	Time	Message
INF	08:17:06	Configuration saved to D:\Temp\config.ecc
INF	08:07:29	EC-Engineer ready. Version 0.9.9

Master/Bus: 1 | Slaves: 17 | Mode: CONFIG | EXPERT

7 Licensing

The license activation can be done on a pc basis with a hardware based license key or by using a USB dongle. A USB dongle license can be used on multiple systems.

If the pc based licensing is used, a hardware ID file has to be generated with the program "Hardware ID File" found in Start->Ackermann Automation->Ackermann Automation->2.x. The generated hardware id file has to be sent to Ackermann Automation to get an activation key, which can be used with the System Open VI. The hardware ID is generated from CPU and Bios information only.



License Number: EcatAcq-20001

Destination Folder Path: c:\temp

8 Getting Started

The best way to get started is by testing and understanding the examples. All needed functions are shown in these examples.

The examples can be found in:
...\Program files\National Instruments\LabVIEW xx\examples\Ackermann Automation\EtherCAT Acquisition

9 Support

For support contact:

Ackermann Automation GmbH
Kelsterbacher Strasse 15-19
60528 Frankfurt am Main
www.ackermann-automation.de

Tel.: +49(0)69-40562742
Fax: +49(0)69-40562816
info@ackermann-automation.de