



acontis technologies GmbH

SOFTWARE

EC-Engineer Layer 5 SDK

User Manual

Version 3.7

Edition: May 4, 2022

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

© Copyright **acontis technologies GmbH**

Neither this document nor excerpts therefrom may be reproduced, transmitted, or conveyed to third parties by any means whatever without the express permission of the publisher. At the time of publication, the functions described in this document and those implemented in the corresponding hardware and/or software were carefully verified; nonetheless, for technical reasons, it cannot be guaranteed that no discrepancies exist. This document will be regularly examined so that corrections can be made in subsequent editions. Note: Although a product may include undocumented features, such features are not considered to be part of the product, and their functionality is therefore not subject to any form of support or guarantee.

Contents

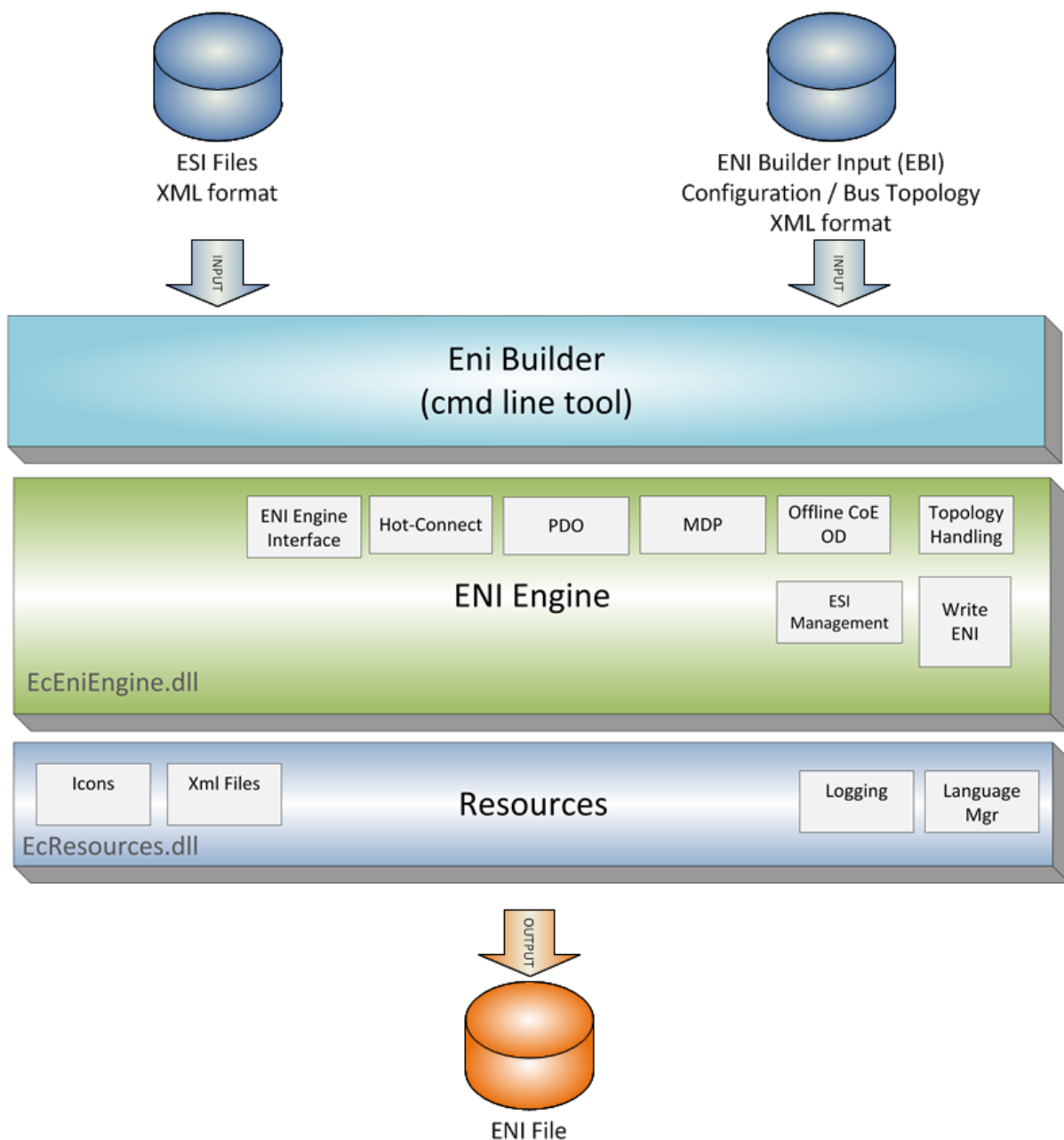
1	Introduction	4
1.1	Overview	4
1.2	Requirements	5
1.3	EtherCAT Slave descriptions (ESI files)	5
2	Tool usage	6
2.1	Microsoft .NET Framework: Windows	6
2.2	Microsoft .NET Framework: Linux	7
2.3	Microsoft .NET Core: Windows	7
2.4	Microsoft .NET Core: Linux	8
3	Project structure	9
4	ENI Builder Input (EBI) file format	10
4.1	Data types	10
4.2	Tags	10
4.3	Create EBI file using EC-Engineer	17
4.4	Type Specific Settings	18
4.5	Examples	20
5	Sample Code for C#	28
5.1	Getting Started	28
5.2	Architecture	28
5.3	Command line interface	29
6	Sample Code for C++	30
6.1	Getting Started	30
6.2	Architecture	30
6.3	Command line interface	31
7	How to integrate EniEngine	32
7.1	Create Environment	32
7.2	Configure created Device	32
7.3	Create slaves	32
7.4	Build Tree and configure slaves	33
7.5	Generate ENI	33

1 Introduction

1.1 Overview

This document describes how the “EC-Engineer SDK EniEngine Layer 5” can be integrated into an already existing software environment by using the “ENI Builder” sample code (for C# or C++).

This document describes the usage of the “ENI Builder”. It is a command line utility which is controlled by a XML based ENI Builder Input (EBI) file (`config.xml`) and generates the EtherCAT bus description file (EtherCAT Network Information, ENI) according to ETG.2100 (v1.0.0). The EBI file describes the EtherCAT bus topology (identity of EtherCAT slave devices and how they are connected to each other) and additional settings of each slave. Further HotConnect / HotPlug feature is supported. In order to generate the ENI, the EtherCAT Slave Information (ESI) according to ETG.2000 (v.1.0.1) is required for each slave.



1.2 Requirements

1.2.1 For Developers

Microsoft Windows 7 and above

Microsoft .NET Framework
Visual Studio 2015

Microsoft .NET Core
Visual Studio 2019

1.2.2 For Deployment

Microsoft .NET Framework

- **Windows (x86/x64)**
 - Microsoft Windows 7 and above
 - Microsoft .NET Framework 4 Client Profile
 - Microsoft Visual C++ 2010 Runtime
- **Linux (x86/x64)**
 - Mono 3.2 and above (install mono-complete)

Microsoft .NET Core

- **Windows (x86/x64)**
 - Microsoft Windows 7 and above
 - Microsoft Visual C++ 2010 Runtime
- **Linux (x64/ARM)**
 - Ubuntu 16.04 x64 and above
- **.NET Core 3.1 Runtime**
 - <https://dotnet.microsoft.com/en-us/download/dotnet>

Memory as recommended for operating system

Disk space approximately 80 MB (depend on number of ESI files)

1.3 EtherCAT Slave descriptions (ESI files)

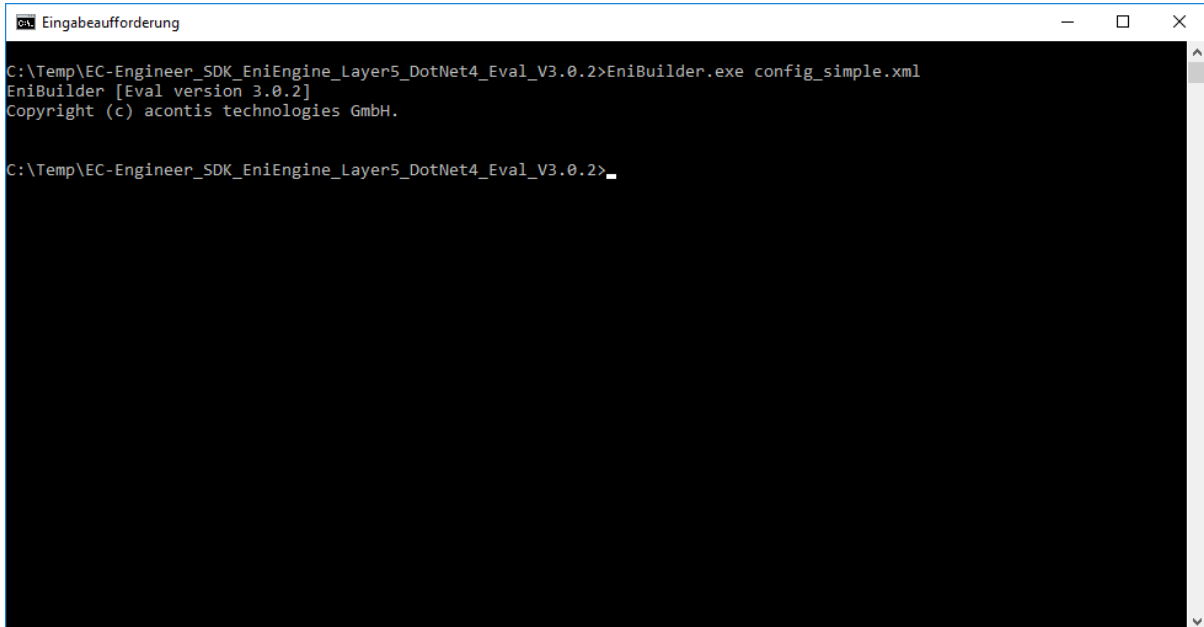
The ESI files of the connected slaves must be located in the “EtherCAT” subdirectory.

Upon first program start, the tool will scan this directory for ESI files and writes out a cache file (ESICache.xml). This file is read in on subsequent program starts in order to shorten the start time.

ESICache.xml must be deleted by the user if new ESI files are copied to the ESI directory in order to recreate the ESI cache on the next program start.

2 Tool usage

2.1 Microsoft .NET Framework: Windows




```
Eingabeaufforderung
C:\Temp\EC-Engineer_SDK_EniEngine_Layer5_DotNet4_Eval_V3.0.2>EniBuilder.exe config_simple.xml
EniBuilder [Eval version 3.0.2]
Copyright (c) acontis technologies GmbH.

C:\Temp\EC-Engineer_SDK_EniEngine_Layer5_DotNet4_Eval_V3.0.2>
```

`EniBuilder.exe [config.xml]`

The first argument is the ENI Builder Input (EBI) file name which should be read in. This argument is optional and defaults to “config.xml”.

2.2 Microsoft .NET Framework: Linux



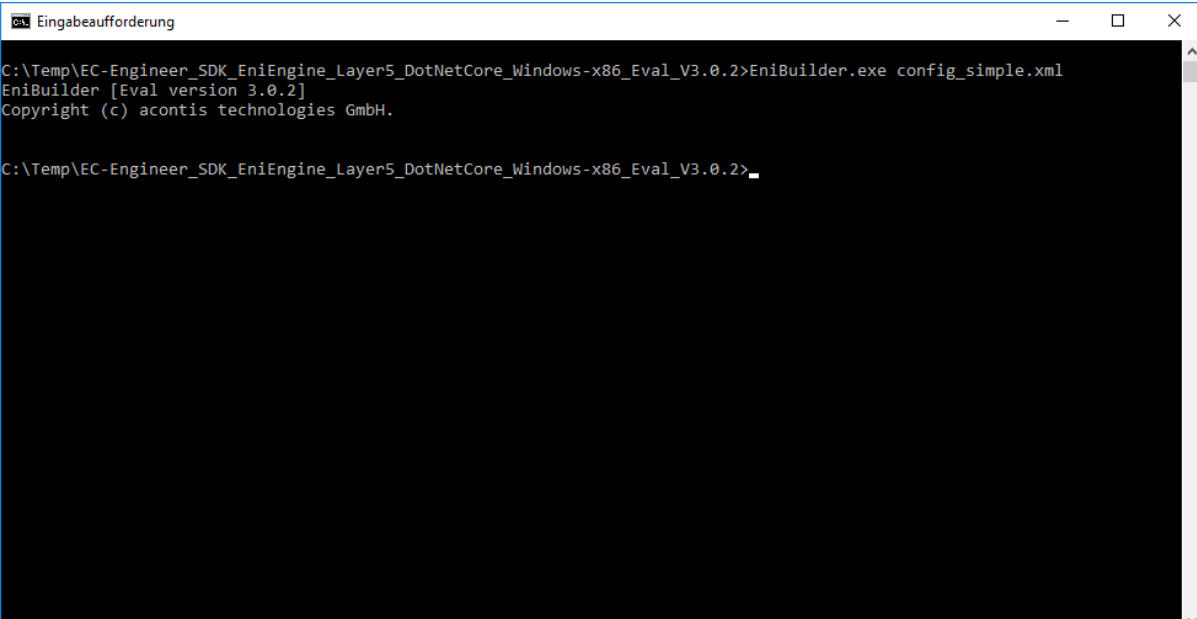
```
rte@ubuntu: ~/EC-Engineer_SDK_EniEngine_Layer5_DotNet4_Eval_V3.0.2
rte@ubuntu:~/EC-Engineer_SDK_EniEngine_Layer5_DotNet4_Eval_V3.0.2$ mono EniBuilder.exe config_simple.xml
EniBuilder [Eval version 3.0.2]
Copyright (c) acontis technologies GmbH.

rte@ubuntu:~/EC-Engineer_SDK_EniEngine_Layer5_DotNet4_Eval_V3.0.2$
```

```
mono EniBuilder.exe [config.xml]
```

The first argument is the ENI Builder Input (EBI) file name which should be read in. This argument is optional and defaults to “config.xml”.

2.3 Microsoft .NET Core: Windows



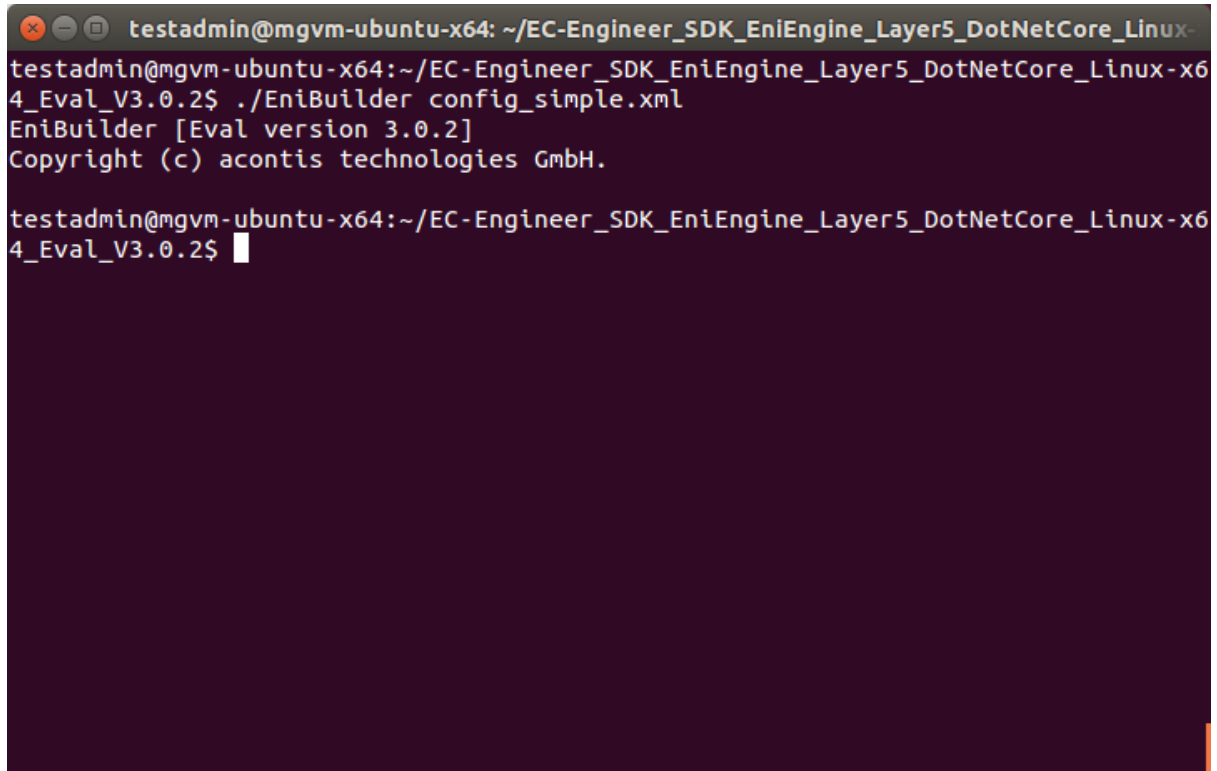
```
Eingabeaufforderung
C:\Temp\EC-Engineer_SDK_EniEngine_Layer5_DotNetCore_Windows-x86_Eval_V3.0.2>EniBuilder.exe config_simple.xml
EniBuilder [Eval version 3.0.2]
Copyright (c) acontis technologies GmbH.

C:\Temp\EC-Engineer_SDK_EniEngine_Layer5_DotNetCore_Windows-x86_Eval_V3.0.2>
```

```
EniBuilder.exe [config.xml]
```

The first argument is the ENI Builder Input (EBI) file name which should be read in. This argument is optional and defaults to “config.xml”.

2.4 Microsoft .NET Core: Linux



```
testadmin@mgvm-ubuntu-x64: ~/EC-Engineer_SDK_EniEngine_Layer5_DotNetCore_Linux-
4_Eval_V3.0.2$ ./EniBuilder config_simple.xml
EniBuilder [Eval version 3.0.2]
Copyright (c) acontis technologies GmbH.

testadmin@mgvm-ubuntu-x64:~/EC-Engineer_SDK_EniEngine_Layer5_DotNetCore_Linux-x6
4_Eval_V3.0.2$
```

```
./EniBuilder [config.xml]
```

The first argument is the ENI Builder Input (EBI) file name which should be read in. This argument is optional and defaults to “config.xml”.

3 Project structure

ENI Sample ENI Files (one for each EBI file)

Languages

Language specific files

EtherCAT

EtherCAT Slave Information (ESI) Files

SrcC#EniBuilder.sln

Visual Studio 2015 Solution File for the C# Sample Code

SrcC#EniBuilderDnc.sln

Visual Studio 2017 Solution File for the C# Sample Code (.NET Core)

SrcCppEniBuilderForCpp.sln

Visual Studio 2015 Solution File for the C++ Sample Code

ESICache.xml (generated on first run)

ESI-File-Cache for faster access of ESI files

NOTE: If you change the ESI-File-Cache, this file must be deleted!

EniBuilder.exe

C# Sample Application

EniBuilderForCpp.exe

C++ Sample Application

EniBuilder.log

Log File

Config*.xml

Sample EBI Files

4 ENI Builder Input (EBI) file format

4.1 Data types

HexDecValue: According to ETG.2000, Ch.6.

String: ASCII string

UInt32: 32 bit unsigned integer in decimal notation

Bool: 0 or 1 or True or False

4.2 Tags

Tag name / Attribute	Description	Mandatory / Optional	Data types
Config	Top-level tag	M	
Info	File metadata	M	
EniFileName	Name of the output file (ENI)	M	String
FileFormatVersion	File version. If this version is <= to the version of the executable, the file is compatible to the executable and can be read in.	M	<Major>.<Minor>, both uint32, decimal
WriteVersionStringToEni	True, to write version string of the EniEngine to the ENI file.	O	Bool

Tag name / Attribute	Description	Mandatory / Optional	Data types
Master	Master settings	O	
<i>Master@Name</i>	Name of master	O	String
Master/CycleTime	Cycle time in microseconds (used for DC)	O	UInt32
Master/Dc	Distributed Clocks settings	O	
<i>Dc@Mode</i>	Mode for controlling the DC time.	O	“BusShift” or “MasterShift” or “ExternalSync”
<i>Dc@SyncWindowMonitoring</i>	SyncWindowMonitoring	O	Bool
<i>Dc@ContinuousRunTimeMeasuring</i>	ContinuousRunTimeMeasuring	O	Bool
<i>Dc@SystemTime64Bit</i>	SystemTime64Bit	O	Bool
Master/Groups	Groups settings	O	
<i>Groups/Entry@Members</i>	Members of group	M	Semicolon separated string with slave addresses, like “1001;1002”
<i>Groups/Entry@Name</i>	Name of group	O	String
<i>Groups/Entry@TaskId</i> Task	Id of group	O	HexDecValue
<i>Groups/Entry@PinInputOffset</i>	Input offset of pinned group	O	HexDecValue

continues on next page

Table 1 – continued from previous page

Tag name / Attribute	Description	Mandatory / Optional	Data types
<i>Groups/Entry@PinOutputOffset</i>	Output offset of pinned group	O	HexDecValue
<i>Groups/Entry@HcIdentificationValue</i>	Identification value for a hot connect group	O	HexDecValue
Master/CyclicTasks	Cyclic task settings	O	
<i>CyclicTasks/Entry@TaskId</i>	Id of cyclic task	M	String
<i>CyclicTasks/Entry@Comment</i>	Comment of cyclic task	O	String
<i>CyclicTasks/Entry@CycleTime</i>	Cycle time of cyclic task (“Task 0” uses “Master/CycleTime”)	O	HexDecValue
Master/MasterSyncUnits	Master Sync Unit settings	O	
<i>MasterSyncUnits/Entry@MsuId</i>	Id of master sync unit	M	HexDecValue
<i>MasterSyncUnits/Entry@TaskId</i>	TaskId of master sync unit	O	String
<i>MasterSyncUnits/Entry@Name</i>	Name of master sync unit	O	String
<i>MasterSyncUnits/Entry@InputOffset</i>	Input offset of master sync unit	O	HexDecValue
<i>MasterSyncUnits/Entry@OutputOffset</i>	Output offset of master sync unit	O	HexDecValue
Master/SyncUnitPairs	Sync Unit Pairs settings	O	
<i>SyncUnitPairs/Entry@MsuId</i>	Master sync unit id	M	HexDecValue
<i>SyncUnitPairs/Entry@PhysAddr</i>	Slave address	M	HexDecValue
<i>SyncUnitPairs/Entry@SlaveSu</i>	Slave sync unit	O	HexDecValue
Master/TraceVariables	Trace variable settings	O	
<i>TraceVariables/Entry@Name</i>	Name of trace variable	M	String
<i>TraceVariables/Entry@DataType</i>	Datatype of trace variable	M	String
<i>TraceVariables/Entry@BitSize</i>	Bit size of trace variable	M	HexDecValue
Master/CopyLinks	Copy link settings	O	
<i>CopyLinks/Entry@TraceVariableIdx</i>	Index of trace variable	M	String
<i>CopyLinks/Entry@DstPhysAddr</i>	EtherCAT slave station address of destination slave (uint16)	M	String
<i>CopyLinks/Entry@DstPdoIndex</i>	Index of destination PDO	M	HexDecValue
<i>CopyLinks/Entry@DstPdoEntryIndex</i>	Index of destination PDO entry	O	HexDecValue
<i>Copy-Links/Entry@DstPdoEntrySubIndex</i>	SubIndex of destination PDO entry	O	HexDecValue
<i>CopyLinks/Entry@DstAliasOffset</i>	Alias offset of destination PDO entry (variable)	O	HexDecValue

Tag name / Attribute	Description	Mandatory / Optional	Data types
Slaves	Slave Settings		
<i>Slaves/Slave</i>	Description of one slave device	O	
<i>Slave@Name</i>	Name of slave (by default it will be generated in the format “Slave_PhysAddr”)	O	String
<i>Slave@PhysAddr</i>	EtherCAT slave station address (uint16)	M	HexDecValue

continues on next page

Table 2 – continued from previous page

Tag name / Attribute	Description	Mandatory / Optional	Data types
<i>Slave/Description</i>	Identification of the slave device in order to lookup the corresponding ESI file.	M	
<i>Description@VendorId</i>	Slave's Vendor ID	M	HexDecValue
<i>Description@ProductCode</i>	Slave's Product Code	M	HexDecValue
<i>Description@RevisionNo</i>	Slave's Revision Number	M	HexDecValue
<i>Slave/PreviousPort</i>	Topology Info. May be omitted if this is the first bus slave or if this is the first slave device of a Hot-Connect group which can be connected anywhere in the bus topology.	O	
<i>PreviousPort/PhysAddr</i>	Station address of predecessor slave	M	HexDecValue, uint16
<i>PreviousPort/Port</i>	Outgoing port of predecessor slave	M	"B", "C" or "D" according to ESI.
Slave/Dc	Distributed Clocks settings	O	
<i>Dc@Id</i>	Selection of DC operation mode. The Id corresponds with the Op-Mode/Name tag in the ESI file.	O	String
Slave/Pdo	PDO settings	O	
<i>Slave/Pdo@Sorting</i>	Flag for sorting PDOs of slave automatically (ascending)	O	Bool
<i>Slave/Pdo/Add</i>	Add PDOs	O	
<i>Add/Entry@Name</i>	Name of Pdo	O	
<i>Add/Entry@Index</i>	Index of Pdo	M	HexDecValue
<i>Add/Entry@Type</i>	Type of Pdo	O	Values: Tx/Rx
<i>Slave/Pdo/Remove</i>	Remove PDOs	O	
<i>Remove/Entry@Index</i>	Index of Pdo	M	HexDecValue
<i>Slave/Pdo/Edit</i>	Edit PDOs	O	
<i>Edit/Entry@Index</i>	Index of Pdo (if SlotNr is used, it must contain the index of the MDP-Module)	M	HexDecValue
<i>Edit/Entry@SlotNr</i>	SlotNr of MDP-Slave	O	HexDecValue
<i>Edit/Entry@Type</i>	Type of Pdo	M	Values: Tx/Rx
<i>Edit/Entry@Name</i>	New name of Pdo	M	
Slave/PdoEntry	PDO entry settings	O	
<i>Slave/PdoEntry/Add</i>	Add PDO Entries	O	
<i>Add/Entry@PdoIndex</i>	Index of Pdo (if SlotNr is used, it must contain the PDO Index of the MDP-Module)	M	HexDecValue
<i>Add/Entry@SlotNr</i>	SlotNr of MDP-Slave	O	HexDecValue
<i>Add/Entry@Name</i>	Name of Pdo Entry	M	
<i>Add/Entry@Comment</i>	Comment for Pdo Entry	O	

continues on next page

Table 2 – continued from previous page

Tag name / Attribute	Description	Mandatory / Optional	Data types
<i>Add/Entry@Index</i>	Index of Pdo Entry (if SlotNr is used, it must contain the index of the MDP-Module)	M	HexDecValue
<i>Add/Entry@SubIndex</i>	SubIndex of Pdo Entry	M	HexDecValue
<i>Add/Entry@DataType</i>	DataType of Pdo Entry	M	e.g. #xINT
<i>Add/Entry@Size</i>	Size of Pdo Entry	M	HexDecValue
<i>Slave/PdoEntry/Remove</i>	Remove PDO Entries	O	
<i>Remove/Entry@PdoIndex</i>	Index of Pdo (if SlotNr is used, it must contain the PDO Index of the MDP-Module)	M	HexDecValue
<i>Remove/Entry@SlotNr</i>	SlotNr of MDP-Slave	O	HexDecValue
<i>Remove/Entry@Index</i>	Index of Pdo Entry (if SlotNr is used, it must contain the index of the MDP-Module)	M	HexDecValue
<i>Remove/Entry@SubIndex</i>	SubIndex of Pdo Entry	M	HexDecValue
<i>Slave/PdoEntry/Edit</i>	Edit PDO Entries	O	
<i>Edit/Entry@PdoIndex</i>	Index of Pdo (if SlotNr is used, it must contain the PDO Index of the MDP-Module)	M	HexDecValue
<i>Edit/Entry@SlotNr</i>	SlotNr of MDP-Slave	O	HexDecValue
<i>Edit/Entry@Index</i>	Index of Pdo Entry (if SlotNr is used, it must contain the index of the MDP-Module)	M	HexDecValue
<i>Edit/Entry@SubIndex</i>	SubIndex of Pdo Entry	M	HexDecValue
<i>Edit/Entry@Name</i>	New name of Pdo Entry	M	
<i>Edit/Entry@Comment</i>	New comment of Pdo Entry	O	
Slave/AliasVariables	PDO entry settings	O	
<i>Slave/AliasVariables/Add</i>	Add Alias Variables	O	
<i>Add/Entry@PdoIndex</i>	Index of Pdo of the alias variable	M	HexDecValue
<i>Add/Entry@PdoEntryIndex</i>	Index of PdoEntry of the alias variable	M	HexDecValue
<i>Add/Entry@PdoEntrySubIndex</i>	SubIndex of PdoEntry of the alias variable	M	
<i>Add/Entry@Direction</i>	Direction of alias variable	M	String: "Input", "Output"
<i>Add/Entry@Offset</i>	Offset of alias variable	M	HexDecValue
<i>Add/Entry@Name</i>	Name of alias variable	M	String
<i>Add/Entry@Size</i>	Size of alias variable	M	HexDecValue
<i>Add/Entry@DataType</i>	Data type of alias variable	O	String
Slave/ExcludePdo	Excludes PDO settings	O	
<i>Slave/ExcludePdo/Add</i>	Excludes PDO	O	
<i>Add/Entry@Index</i>	Index of PDO to exclude (if SlotNr is used, it must contain the PDO Index of the MDP-Module)	M	HexDecValue
<i>Add/Entry@SlotNr</i>	SlotNr of MDP-Slave	O	HexDecValue
<i>Slave/ExcludePdo/Remove</i>	Includes PDO	O	

continues on next page

Table 2 – continued from previous page

Tag name / Attribute	Description	Mandatory / Optional	Data types
<i>Remove/Entry@Index</i>	Index of PDO to include (if SlotNr is used, it must contain the PDO Index of the MDP-Module)	M	HexDecValue
<i>Remove/Entry@SlotNr</i>	SlotNr of MDP-Slave	O	HexDecValue
<i>Remove/Entry@SyncManager</i>	Index of SyncManager	O	HexDecValue
Slave/InitCmd	CoE InitCmd settings	O	
<i>Slave/InitCmd/Add</i>	Add CoE InitCmds	O	
<i>Add/Entry@Index</i>	Index of InitCmd	M	HexDecValue
<i>Add/Entry@SubIndex</i>	SubIndex of InitCmd	M	HexDecValue
<i>Add/Entry@Ccs</i>	Direction of InitCmd	M	1 =Download; 2 = Upload
<i>Add/Entry@Transitions</i>	Transitions of InitCmd (comma separated)	M	
<i>Add/Entry@Data</i>	Data of InitCmd	O	
<i>Add/Entry@Comment</i>	Comment of InitCmd	O	
<i>Add/Entry@CompleteAccess</i>	CompleteAccess of InitCmd	O	Bool
<i>Slave/InitCmd/Remove</i>	Remove CoE InitCmds	O	
<i>Remove/Entry@Index</i>	Index of InitCmd (if SlotNr is used, it must contain the PDO Index of the MDP-Module)	M	HexDecValue
<i>Remove/Entry@SlotNr</i>	SlotNr of MDP-Slave	O	HexDecValue
<i>Remove/Entry@SubIndex</i>	SubIndex of InitCmd	M	HexDecValue
<i>Remove/Entry@Ccs</i>	Direction of InitCmd	M	1 =Download; 2 = Upload
<i>Slave/InitCmd/Edit</i>	Edit CoE InitCmds	O	
<i>Edit/Entry@Index</i>	Index of InitCmd (if SlotNr is used, it must contain the PDO Index of the MDP-Module)	M	HexDecValue
<i>Edit/Entry@SlotNr</i>	SlotNr of MDP-Slave	O	HexDecValue
<i>Edit/Entry@SubIndex</i>	SubIndex of InitCmd	M	HexDecValue
<i>Edit/Entry@Ccs</i>	Direction of InitCmd	M	1 =Download; 2 = Upload
<i>Edit/Entry@Data</i>	New data of InitCmd	M	
Slave/SoeInitCmd	SoE InitCmd settings	O	
<i>Slave/SoeInitCmd/Add</i>	Add SoE InitCmds	O	
<i>Add/Entry@Idn</i>	IDN of InitCmd	M	HexDecValue
<i>Add/Entry@Chn</i>	Channel of InitCmd	M	HexDecValue
<i>Add/Entry@Transitions</i>	Transitions of InitCmd (comma separated)	M	
<i>Add/Entry@Data</i>	Data of InitCmd	O	
<i>Add/Entry@Comment</i>	Comment of InitCmd	O	
<i>Slave/SoeInitCmd/Remove</i>	Remove SoE InitCmds	O	
<i>Add/Entry@Idn</i>	IDN of InitCmd	M	HexDecValue
<i>Add/Entry@Chn</i>	Channel of InitCmd	M	HexDecValue
<i>Slave/SoeInitCmd/Edit</i>	Edit SoE InitCmds	O	
<i>Edit/Entry@Idn</i>	IDN of InitCmd	M	HexDecValue
<i>Edit/Entry@Chn</i>	Channel of InitCmd	M	HexDecValue
<i>Edit/Entry@Data</i>	New data of InitCmd	M	
Slave/Mdp	MDP settings	O	
<i>Slave/Mdp/Add</i>	Assign MDP-Module	O	
<i>Add/Entry@SlotNr</i>	SlotNr of MDP-Slave	M	HexDecValue

continues on next page

Table 2 – continued from previous page

Tag name / Attribute	Description	Mandatory / Optional	Data types
<i>Add/Entry@ModuleIdent</i>	ModuleIdent of MDP-Module	M	HexDecValue
<i>Add/Entry@Name</i>	Name of Slot/Module-Pair	O	String
<i>Slave/Mdp/Edit</i>	Edit MDP-Module	O	
<i>Edit/Entry@SlotNr</i>	SlotNr of MDP-Slave	M	HexDecValue
<i>Edit/Entry@Name</i>	Name of Slot/Module-Pair	M	String
<i>Slave/Mdp/Remove</i>	Not assign MDP-Slot	O	
<i>Remove/Entry@SlotNr</i>	SlotNr of MDP-Slave	M	HexDecValue
Slave/CopyLink	Slave to Slave copy	O	
<i>Slave/CopyLink/Add</i>	Add “CopyLink”	O	
<i>Add/Entry@PhysAddr</i>	EtherCAT slave station address of destination slave (uint16)	M	HexDecValue
<i>Add/Entry@BitSize</i>	Size to copy (bits)	M	HexDecValue
<i>Add/Entry@SrcBitOffs</i>	Offset in source slave (bits)	O	HexDecValue
<i>Add/Entry@DstBitOffs</i>	Offset in destination slave (bits)	O	HexDecValue
<i>Add/Entry@SrcPdoIndex</i>	Index of source PDO	O	HexDecValue
<i>Add/Entry@SrcPdoEntryIndex</i>	Index of source PDO entry	O	HexDecValue
<i>Add/Entry@SrcPdoEntrySubIndex</i>	SubIndex of source PDO entry	O	HexDecValue
<i>Add/Entry@SrcAliasOffset</i>	Alias offset of source PDO entry (variable)	O	HexDecValue
<i>Add/Entry@DstPdoIndex</i>	Index of destination PDO	O	HexDecValue
<i>Add/Entry@DstPdoEntryIndex</i>	Index of destination PDO entry	O	HexDecValue
<i>Add/Entry@DstPdoEntrySubIndex</i>	SubIndex of destination PDO entry	O	HexDecValue
<i>Add/Entry@DstAliasOffset</i>	Alias offset of destination PDO entry (variable)	O	HexDecValue
<i>Add/Entry@InCycle</i>	Enable S2S copy link in one cycle	O	Bool
Slave/EoE	EoE settings	O	
<i>EoE/Enable</i>	Enables EoE	O	Bool
<i>EoE/VirtualMacAddress</i>	Virtual MAC address	O	MAC Address (e.g. “02 00 00 00 03 E9” or “Auto” = MAC Address will be generated)
<i>EoE/TimeStampRequested</i>	Time Stamp Requested	O	Bool
<i>EoE/PortMode</i>	Port Mode	O	Bool (True = “IP Port”, False = “Switch Port”)
<i>EoE/OverwriteIpSettings</i>	Overwrite IP Settings	O	Bool
<i>EoE/IpAddress</i>	IP Address	O	IP Address (e.g. “127.0.0.1”)
<i>EoE/SubnetMask</i>	Subnetmask	O	IP Address
<i>EoE/DefaultGateway</i>	Default Gateway	O	IP Address
<i>EoE/DnsServer</i>	DNS Server	O	IP Address
<i>EoE/DnsName</i>	DNS Name	O	String
<i>Slave/Settings</i>	Advanced Settings	O	

continues on next page

Table 2 – continued from previous page

Tag name / Attribute	Description	Mandatory / Optional	Data types
<i>Settings/DisableLRW</i>	Disables LRW	O	Bool
<i>Settings/CheckVendorId</i>	Enables checking vendor ID	O	Bool
<i>Settings/CheckProductCode</i>	Enables checking product code	O	Bool
<i>Settings/CheckRevisionNo</i>	Checking revision number	O	NONE, EQ, EQ_OR_G, LW_EQ, LW_EQ_HW_EQ_OR_G, HW_EQ, HW_EQ_LW_EQ_OR_G
<i>Settings/CheckSerialNo</i>	Enables checking serial number	O	Bool
<i>Settings/IdentificationAdo</i>	Overwrites identification ADO value	O	HexDecValue
<i>Settings/WatchdogMultiplier</i>	Sets watchdog multiplier	O	HexDecValue
<i>Settings/PdiWatchdog</i>	Sets PDI watchdog	O	HexDecValue
<i>Settings/SmWatchdog</i>	Sets SM watchdog	O	HexDecValue
<i>Settings/PotentialRefClock</i>	Enables potential reference clock	O	Bool
<i>Settings/MailboxAccessTimeout</i>	Sets timeout for mailbox access	O	HexDecValue
<i>Settings/PreopTimeout</i>	Sets the timeout for changing state machine from Init to Pre-Op and from Init to Bootstrap	O	HexDecValue
<i>Settings/SafeopOpTimeout</i>	Sets the timeout for changing state machine from Pre-Op to Safe-Op and from Safe-Op to Op	O	HexDecValue
<i>Settings/BackToInitTimeout</i>	Sets the timeout for changing state machine back to Pre-Op or Init	O	HexDecValue
<i>Settings/BackToSafeopTimeout</i>	Sets the timeout for changing state machine from Op to Safe-Op	O	HexDecValue
<i>Settings/MailBoxOutputSize</i>	Sets mailbox input size	O	HexDecValue
<i>Settings/MailBoxInputSize</i>	Sets mailbox output size	O	HexDecValue
Slave/TypeSpecific	Type specific data	O	
<i>TypeSpecific/Settings</i>	Settings for “TypeSpecific”	O	Please refer to “ <i>Type Specific Settings</i> ”
<i>Settings/Entry@Name</i>	Name of setting	M	String (e.g. “DP Slave Parameter Set” for EL6731-0010)
<i>Settings/Entry@SubName</i>	SubName of setting	M	String (e.g. “Station Address” for EL6731-0010)
<i>Settings/Entry@Value</i>	Value of setting	M	String
<i>TypeSpecific/Inputs</i>	Inputs of “TypeSpecific”	O	
<i>Inputs/Entry@Name</i>	Name of input	M	String (e.g. Variable n)
<i>Inputs/Entry@Type</i>	Type of input	M	String (e.g. 1 BYTE, 2 WORD, ...)
<i>TypeSpecific/Outputs</i>	Outputs of “TypeSpecific”	O	
<i>Outputs/Entry@Name</i>	Name of output	M	String (e.g. Variable n)

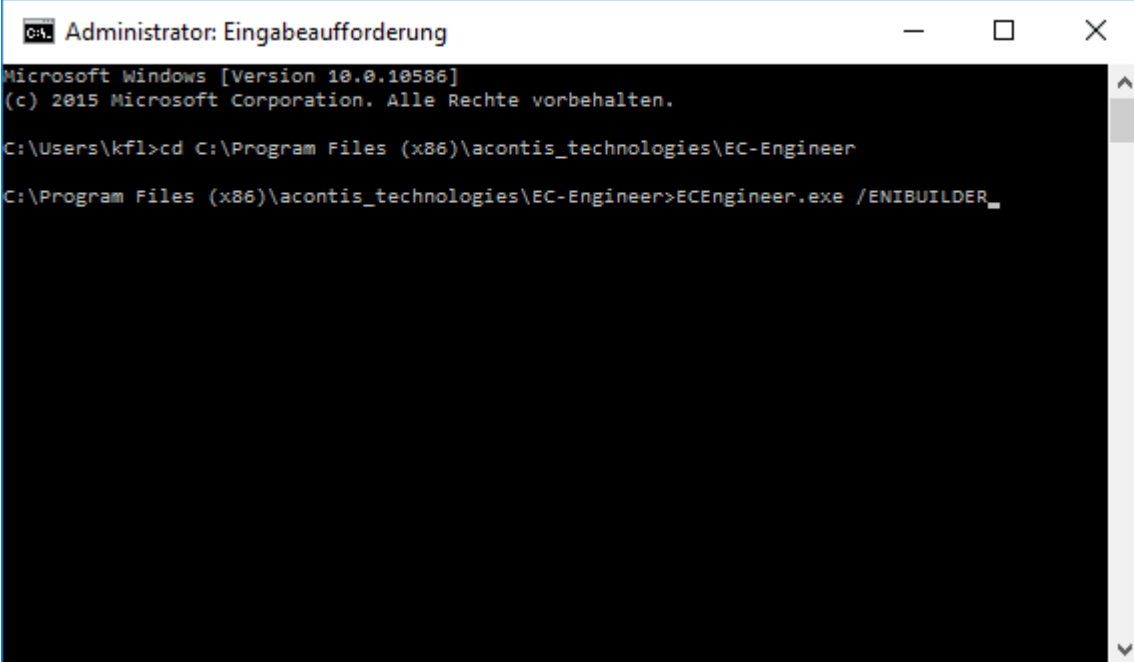
continues on next page

Table 2 – continued from previous page

Tag name / Attribute	Description	Mandatory / Optional	Data types
<i>Outputs/Entry@Type</i>	Type of output	M	String (e.g. 1 BYTE, 2 WORD, ...)

4.3 Create EBI file using EC-Engineer

Instead of creating the EBI file manual, EC-Engineer can be used to create an EBI file. To do that, EC-Engineer must be started with the command line parameter /ENIBUILDER (EC-Engineer User Manual, Chapter 8 Command Line Interface)



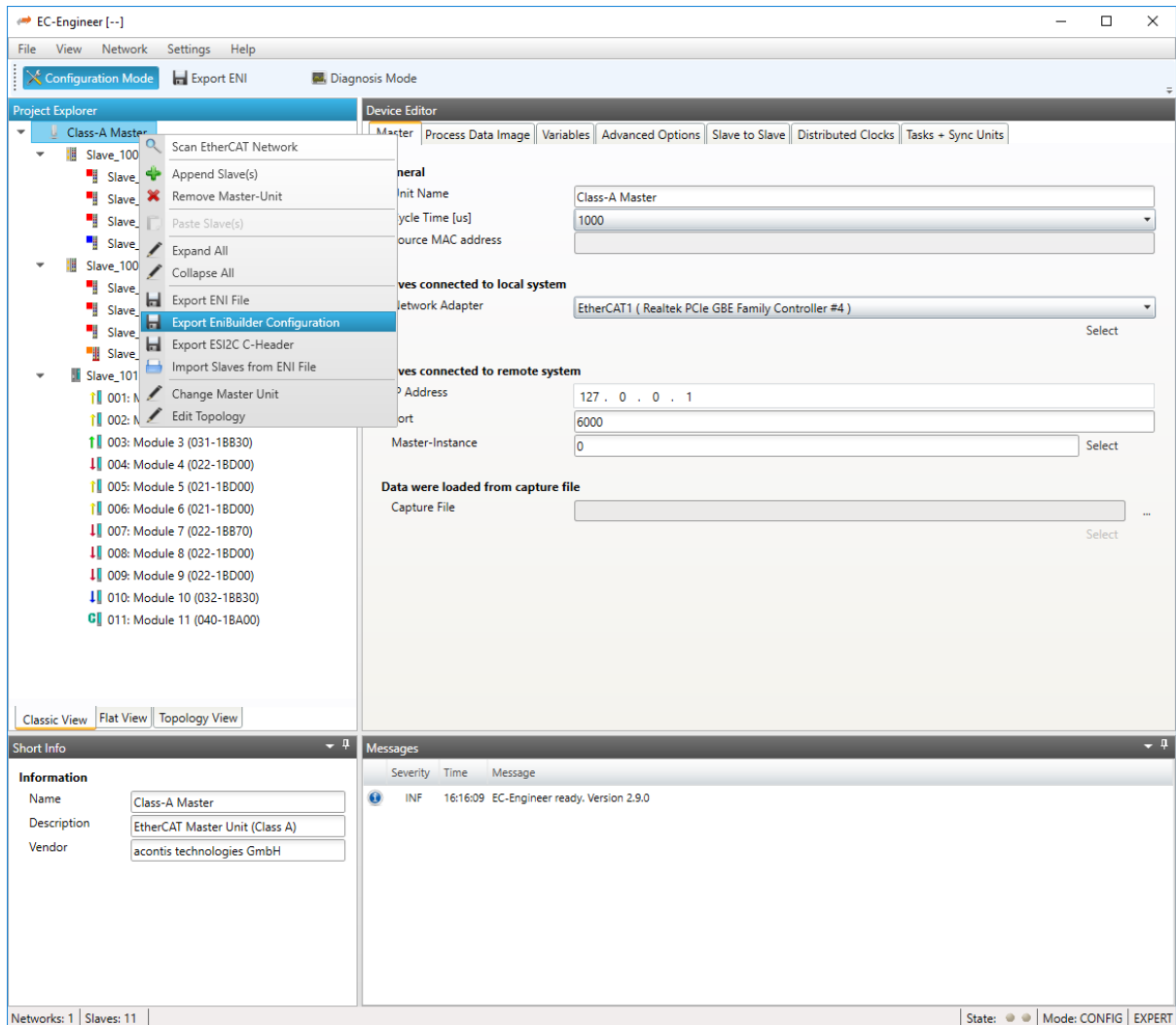
```

Administrator: Eingabeaufforderung
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\kfl>cd C:\Program Files (x86)\acontis_technologies\EC-Engineer
C:\Program Files (x86)\acontis_technologies\EC-Engineer>ECEngineer.exe /ENIBUILDER_

```

If EC-Engineer is started like this, there will be a new entry in the context menu:



So now it is possible to create the whole configuration in the EC-Engineer and then use the context menu to automatically create the EBI file.

4.4 Type Specific Settings

Some slaves need special configuration options and support only special types of inputs and outputs. For more information please refer the manual of EC-Engineer.

4.4.1 EL6731-0010 PROFIBUS DP Slave

Further options:

General

Activate: Activates the automatism for generating PDOs and Init Commands

DP Slave Parameter Set

Station Address: DP station address of the DP slave (permitted values: 0-125) Device Type: DP Ident Number of the DP slave

4.4.2 EL6631-0010 PROFINET IO Device

Further options:

General

Activate: Activates the automatism for generating PDOs and Init Commands

4.4.3 K-bus Coupler / IP Link Coupler

Supported devices:

K-bus Coupler

- BK1120
- BK1150
- BK1250

IP Link Coupler

- IL2300-B110
- IL2301-B110
- IL2302-B110

Further options:

General

Activate: Activates the automatism for generating PDOs and Init Commands
Check Terminals at Startup: Activates the automatism for checking terminals at startup

Terminal N

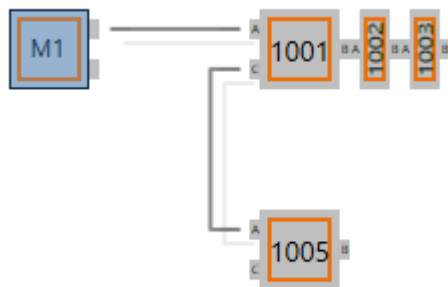
Type: Type of the terminal (see "TcTerminals210.xml")
SubType: SubType of the terminal (see "TcTerminals210.xml")

```

TcTerminals210.xml - Editor
Datei Bearbeiten Format Ansicht ?
</terminal>
<terminal ver="1">
  <type>1002</type>
  <subType>0</subType>
  <abbreviation>KL1002</abbreviation>
  <name LcId="1033">KL 1002, 2 Ch. Input (24V, 3.0ms)</name>
  <name LcId="1031">KL 1002, 2 K. Eingang (24V, 3.0ms)</name>
  <bitTerm>1</bitTerm>
  <group>k1_dig_in</group>
  <channelCount>2</channelCount>
  <fieldbus>
    <lightbus>16</lightbus>
    <profibus>16</profibus>
    <interbus>16</interbus>
    <serial>16</serial>
    <canopen>16</canopen>
    <devicenet>16</devicenet>
    <cp-kbus>16</cp-kbus>
    <ethernet>16</ethernet>
    <sercos>16</sercos>
    <usb>16</usb>
  </fieldbus>
  <subTypeBC>10</subTypeBC>
  <groupBC>k1bc_dig_in</groupBC>
  <channelBC>16</channelBC>
</terminal>
<terminal ver="1">
  <type>9900</type>
  <subType>62</subType>
  
```

4.5 Examples

4.5.1 Example 1: Simple EtherCAT topology



The EBI file for this bus topology looks like this (config_simple.xml):

Tree View	XSL Output
<ul style="list-style-type: none"> xml <ul style="list-style-type: none"> Config <ul style="list-style-type: none"> Info <ul style="list-style-type: none"> EniFileName FileFormatVersion Master <ul style="list-style-type: none"> Name CycleTime Dc Slaves <ul style="list-style-type: none"> Slave <ul style="list-style-type: none"> Name PhysAddr #comment Description Slave <ul style="list-style-type: none"> Name PhysAddr #comment Description PreviousPort <ul style="list-style-type: none"> PhysAddr Port Slave <ul style="list-style-type: none"> Name PhysAddr #comment Description PreviousPort <ul style="list-style-type: none"> PhysAddr Port Slave <ul style="list-style-type: none"> Name PhysAddr #comment Description PreviousPort <ul style="list-style-type: none"> PhysAddr Port 	<pre> version="1.0" encoding="utf-8" ENI/eni_simple.xml 1.1 Class-A Master 1000 Slave_1001 [EK1100] 1001 Slave 1001 (EK1100 EtherCAT Coupler (2A E-Bus)) Slave_1002 [EL2004] 1002 Slave 1002 (EL2004 4Ch. Dig. Output 24V, 0.5A) 1001 B Slave_1003 [EL1004] 1003 Slave 1003 (EL1004 4Ch. Dig. Input 24V, 3ms) 1002 B Slave_1004 [EK1100] 1004 Slave 1004 (EK1100 EtherCAT Coupler (2A E-Bus)) 1001 C </pre>

4.5.2 Example 2: EtherCAT topology with HotConnect

This example has one HotConnect group with two members (EK1100 and EL2008). Note that the head of the Hot-Connect group (EK1100, PhysAddr 1007) has no PreviousPort tag defined, so this group can be connected anywhere in the topology. E.g. to EK1110, PhysAddr 1004, PortB or EK1100, PhysAddr 1001, PortC.

Note: If the PreviousPort tag is defined, the HotConnect group can be connected only to that port specified by the PreviousPort tag (config_hc.xml).

4.5.4 Example 4: PDO exclusion

This configuration fragment shows how to exclude PDO's and PDO-Entries. The slave "SGDV – E1 CoE Drive" has the PDO 0x1a01 (2nd Transmit PDO mapping) defined as default PDO by the ESI (Default PDO: Actually PDO's with assigned Sync-Managers).

In the EBI file, the PDO 0x1a01 (2nd Transmit PDO mapping) will be now excluded and exclusion of PDO 0x1a00 (1st Transmit PDO mapping) will be removed. This means that the PDO assignment will be changed from 0x1a01 to 0x1a00.

Note that the ENI Builder doesn't check if the PDO- or PDO-Entries exclusion is allowed by the ESI description. I.e. digital IO clamps without microcontroller (Device-Emulation) might have fixed PDO's, so the PDO's exclusion is not allowed (not supported by HW) (config_full.xml).

The screenshot displays the ENI Builder interface. On the left, a 'Tree View' shows the configuration structure for an XML file. The path is: Config > Slaves > Slave > ExcludePdo > Add > Entry > Index. The 'Index' property is set to '#x1a01'. Below this, another 'Index' property is set to '#x1a00'. On the right, the 'XSL Output' shows the resulting XML code. It includes a header with 'version="1.0" encoding="utf-8"', followed by the slave name 'Slave_1007 [SGDV-E1 CoE Drive]' and ID '1007'. A comment line reads 'Slave 1007 (SGDV-E1 EtherCAT (CoE) SERVOPACK Rev3)'. The output then shows the '1001' and 'C' values, and finally the 'ExcludePdo' entries for '#x1a01' and '#x1a00'.

4.5.5 Example 5: Configure special slaves

This example shows how to configure special slaves which uses the tag "Type Specific Settings" (config_special.xml)...

Slave 1002: PROFIBUS DP Slave “EL6731-0010” with 1 BYTE IN & OUT

Tree View	XSL Output
<ul style="list-style-type: none"> xml <ul style="list-style-type: none"> Config <ul style="list-style-type: none"> Info Master Slaves <ul style="list-style-type: none"> Slave <ul style="list-style-type: none"> Slave <ul style="list-style-type: none"> Name PhysAddr #comment Description PreviousPort TypeSpecific <ul style="list-style-type: none"> Settings <ul style="list-style-type: none"> Entry <ul style="list-style-type: none"> Name SubName Value Inputs <ul style="list-style-type: none"> Entry <ul style="list-style-type: none"> Name Type Outputs <ul style="list-style-type: none"> Entry <ul style="list-style-type: none"> Name Type Settings Slave Slave 	<pre> version="1.0" encoding="utf-8" Slave_1002 [EL6731-0010] 1002 Slave 1002 (EL6731-0010 PROFIBUS DP Slave) General Activate 1 Variable 0 1 BYTE Variable 0 1 BYTE </pre>

Slave 1003: PROFINET IO Device “EL6631-0010” with 1 BYTE IN & OUT

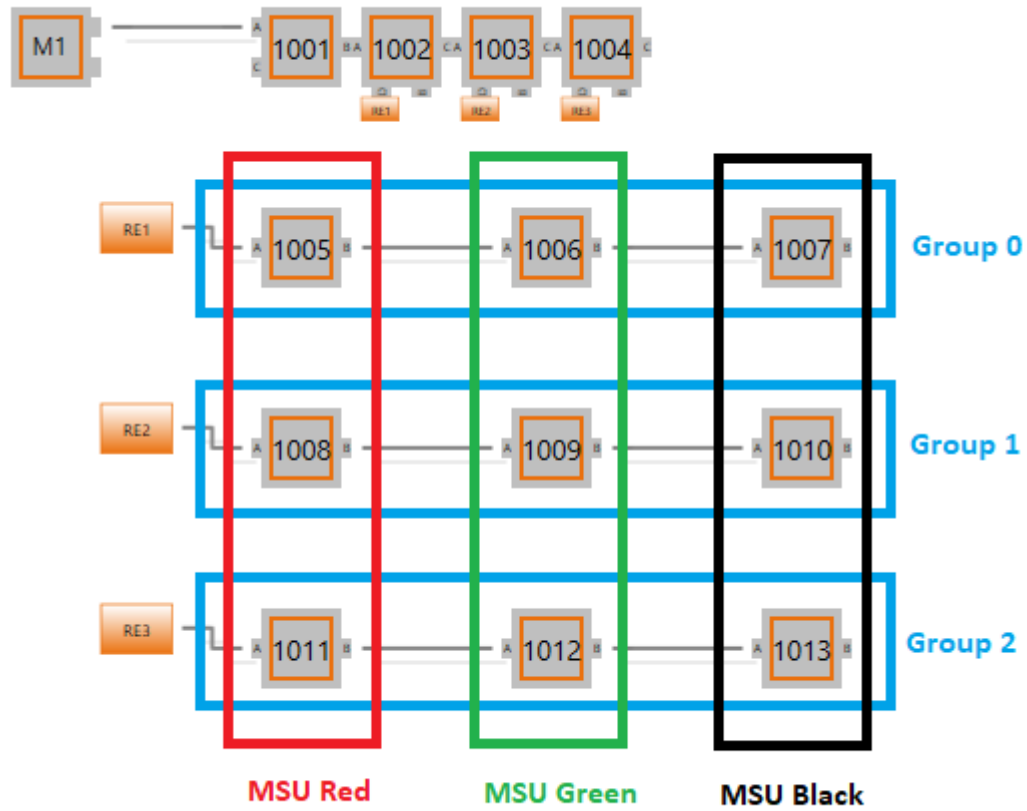
Tree View	XSL Output
<ul style="list-style-type: none"> xml <ul style="list-style-type: none"> Config <ul style="list-style-type: none"> Info Master Slaves <ul style="list-style-type: none"> Slave <ul style="list-style-type: none"> Slave <ul style="list-style-type: none"> Slave <ul style="list-style-type: none"> Name PhysAddr #comment Description PreviousPort TypeSpecific <ul style="list-style-type: none"> Settings <ul style="list-style-type: none"> Entry <ul style="list-style-type: none"> Name SubName Value Inputs <ul style="list-style-type: none"> Entry <ul style="list-style-type: none"> Name Type Outputs <ul style="list-style-type: none"> Entry <ul style="list-style-type: none"> Name Type Settings Slave 	<pre> version="1.0" encoding="utf-8" Slave_1003 [EL6631-0010] 1003 Slave 1003 (EL6631-0010 PROFINET IO Device) General Activate 1 Variable 0 1 BYTE Variable 0 1 BYTE </pre>

Slave 1003: K-bus Coupler “BK1120” with terminals “KL1012” and “KL2012”

Tree View	XSL Output
<ul style="list-style-type: none"> xml <ul style="list-style-type: none"> Config <ul style="list-style-type: none"> Info Master Slaves <ul style="list-style-type: none"> Slave <ul style="list-style-type: none"> Slave Slave Slave Slave 	<pre> version="1.0" encoding="utf-8" Slave_1004 [BK1120] 1004 Slave 1004 (BK1120 EtherCAT Fieldbus coupler) General Activate 1 Terminal 0 Type 1012 Terminal 1 Type 2012 </pre>

4.5.6 Example 6: Master Sync Units

This example shows how to configure Master Sync Units (config_msu.xml).



The EBI file for this bus topology looks like this (`config_msu.xml`):

Tree View	XSL Output
<ul style="list-style-type: none"> xml <ul style="list-style-type: none"> Config <ul style="list-style-type: none"> Info Master <ul style="list-style-type: none"> Name CycleTime Dc <ul style="list-style-type: none"> Groups <ul style="list-style-type: none"> Entry <ul style="list-style-type: none"> Members: 1005;1006;1007 Name: Group 0 TaskId: 0 HcIdentificationValue: 1005 Entry <ul style="list-style-type: none"> Members: 1008;1009;1010 Name: Group 1 TaskId: 0 HcIdentificationValue: 1008 Entry <ul style="list-style-type: none"> Members: 1011;1012;1013 Name: Group 2 TaskId: 0 HcIdentificationValue: 1011 MasterSyncUnits <ul style="list-style-type: none"> Entry <ul style="list-style-type: none"> MsuId: 1000 TaskId: 0 Name: MasterSyncUnit Red Entry Entry SyncUnitPairs <ul style="list-style-type: none"> Entry <ul style="list-style-type: none"> MsuId: 1000 PhysAddr: 1005 SlaveSu: 0 Entry Entry Entry Entry Entry Entry Entry 	<pre> version="1.0" encoding="utf-8" Class-A Master 1000 1005;1006;1007 Group 0 0 1005 1008;1009;1010 Group 1 0 1008 1011;1012;1013 Group 2 0 1011 1000 0 MasterSyncUnit Red 1000 1005 0 </pre>

5 Sample Code for C#

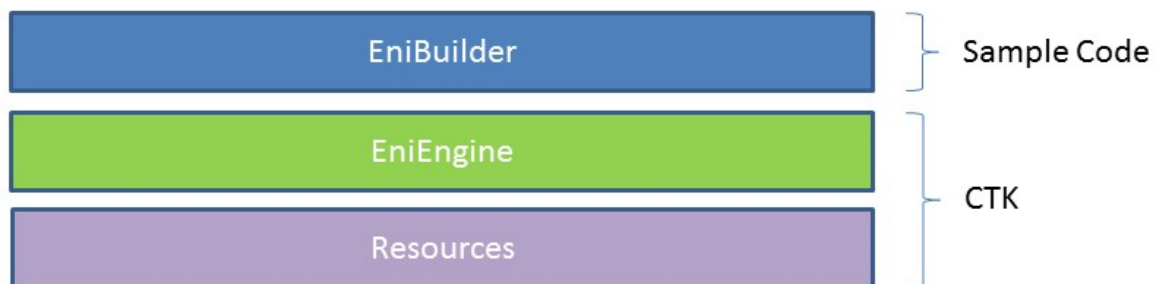
5.1 Getting Started

Open the Visual Studio 2015 Solution (“SrcC#EniBuilder.sln”) or Visual Studio 2017 Solution (“SrcC#EniBuilderDnc.sln”) and do the following things:

- Compile the solution (Output folder of the “Debug” configuration is the root folder, where all the binaries are located)
- Run `EniBuilder.exe` with parameter “C:myEniBuilderDirconfig_simple.xml”, which will generate an ENI file based on the EBI file settings

5.2 Architecture

ENI Builder contains the following components:



1. Sample Code

EniBuilder.exe

- C# Console Application, which parses the command line parameters and loads the “EniEngine” for generating the ENI file.

2. CTK

EniEngine (EcEniEngine.dll)

- EtherCAT Slave Information File (ESI)
- EtherCAT Network Information File (ENI)
- ENI Generation (PDO, MDP, CoE, DC, Hot Connect, ...)
- EniBuilder Input File Format (.ebi)

Resources (EcResources.dll)

- Language Manager
- Logger

5.3 Command line interface

The command line interface supports the following commands:

Name of the EBI file

Specify the path to the EBI file which contains all information for generating the ENI file

/APPDATA = "Path to AppData directory" (optional)

Specify the path to the AppData directory. It contains a subfolder "EtherCAT" with all ESI files and it is also the path for the logfile.

/INDENTXMLFILES (optional)

Activates indenting of XML files (makes exported XML files readable, but increases size).

/NOEBISHEMA (optional)

Deactivates EBI schema check (maybe necessary for compatibility reasons).

/IGNOREREVISION (optional)

Deactivates the revision check, if slave was not found and tries to find a compatible slave.

/LOGLEVEL ="ERR" (optional)

Specify the log level (possible values are: DBG = Debug, INF = Info, WRN = Warning, ERR = Error)

/PROCESSIMAGELAYOUT (optional)

For flags please see EC-Engineer documentation. Please enter decimal values.

/ALLOWDUPLICATENAMES (optional)

Allows duplicate names e.g. for slaves and variables.

6 Sample Code for C++

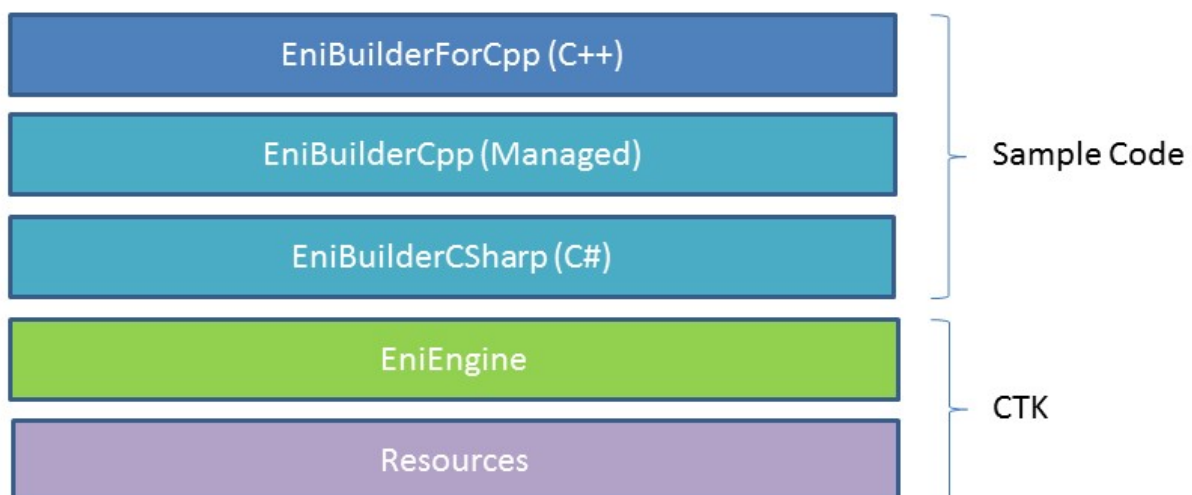
6.1 Getting Started

Open the Visual Studio 2015 Solution (“SrcCppEniBuilderForCpp.sln”) and do the following things:

- Remove unavailable projects (EcResources, EcEniEngine, ...)
- Compile the solution (Output folder of the “Debug” configuration is the root folder, where all the binaries are located)
- Run “EniBuilderForCpp.exe” with parameter “C:myEniBuilderDirconfig_simple.xml”, which will generate an ENI file based on the EBI file settings

6.2 Architecture

EniBuilderForCpp contains the following components:



1. Sample Code

EniBuilderForCpp.exe

- Win32 Console Application, which parses the command line parameters and loads the EniBuilderCpp.dll

EniBuilderCpp.dll

- Win32 Managed Code Library, which exports a wrapper class named “CEniBuilderCpp”. This class converts the data from C++ to C# and back

EniBuilderCSharp.dll

- C# Class Libaray, which loads the “EniEngine” for generating the ENI file

2. CTK

EniEngine (EcEniEngine.dll)

- EtherCAT Slave Information File (ESI)
- EtherCAT Network Information File (ENI)
- ENI Generation (PDO, MDP, CoE, DC, Hot Connect, ...)
- EniBuilder Input File Format (.ebi)

Resources (EcResources.dll)

- Language Manager
- Logger

6.3 Command line interface

The command line interface supports the following commands:

Name of the EBI file

Specify the path to the EBI file which contains all information for generating the ENI file.

/APPDATA =”Path to AppData directory” (optional)

Specify the path to the AppData directory. It contains a subfolder “EtherCAT” with all ESI files and it is also the path for the logfile.

/INDENTXMLFILES (optional)

Activates indenting of XML files (makes exported XML files readable, but increases size).

/IGNOREREVISION (optional)

Deactivates the revision check, if slave was not found and tries to find a compatible slave.

7 How to integrate EniEngine

This chapter gives you an idea how to integrate the EniEngine.

7.1 Create Environment

At first you need to create the environment. You need for example the path to the ESI files and have to create a `EniFile` object.

```
var defaultFileIo = new EcDefaultFileIo
{
    EsiPath = "PathToESIs" //Path to your ESI files
};

// Initialize ENI Engine's factory.
EniFile = new EcEniFile
(null, // optional implement IErrorHandler for Error handling (see below)
defaultFileIo);

// Load ESI cache (ESICache.xml) or create a new one if not already existing.
EniFile.EsiManager.LoadCache();
```

If you need Error Messages for the Error Handler, implement this function.

```
public void HandleEniEngineNotification(EcEniEngineCode code, params object[] args)
```

Conclusion: Now a `EcEniFile` object with an `EcEniDevice` is created.

7.2 Configure created Device

Now you can pick your device and change name and settings.

```
var device = EniFile.Device;

device.Name = "MyMasterName";
device.SettingsData.CycleTimeUs = 1000;
device.SettingsUpdate(device.SettingsData); //Important to update the value!!
```

7.3 Create slaves

Now it is possible to create some slaves:

```
var ek1100 = device.CreateSlave(2, 72100946, 1179648); //(VendorId, ProductCode, ↵
↵RevisionNumber)
var el2004 = device.CreateSlave(2, 131346514, 1179648);
var el3152 = device.CreateSlave(2, 206581842, 1245184);
```


7.4 Build Tree and configure slaves

Now that there is a device and some slaves you have to build your tree.

```
device.AddSlave(ek1100);
ek1100.AppendSlave(e12004, EcEniSlave.SlavePort.PortName.EPortB);
e12004.AppendSlave(e13152, EcEniSlave.SlavePort.PortName.EPortB);
```

Also you can change settings of the slaves.

```
ek1100.SettingsData.Settings.CheckRevisionNo = EcEniSlaveSettings.CheckType.ENone;
ek1100.SettingsData.Write(ek1100.SettingsData.Settings); //Important to update the
↔value!!
```

You can also change other things, like adding a CoE InitCommand if CoE is supported

```
if (e13152.IsSupported(EcEniSlave.EFeature.ECoe))
{
    var coe = new EcEniSlaveCoe.InitCmd { Ccs = 1, Index = 9999, SubIndex = 0 }
↔;
    coe.Transition.Add(EcEniSlaveInitCmd.ETransition.EOp);
    e13152.CoeData.AddInitCmd(coe);
}
```

7.5 Generate ENI

If you have changed all the things you like, you can generate the ENI file:

```
device.GenerateEni("PathToENIFile");
```

So you generated an ENI file with less than 100 lines of code.