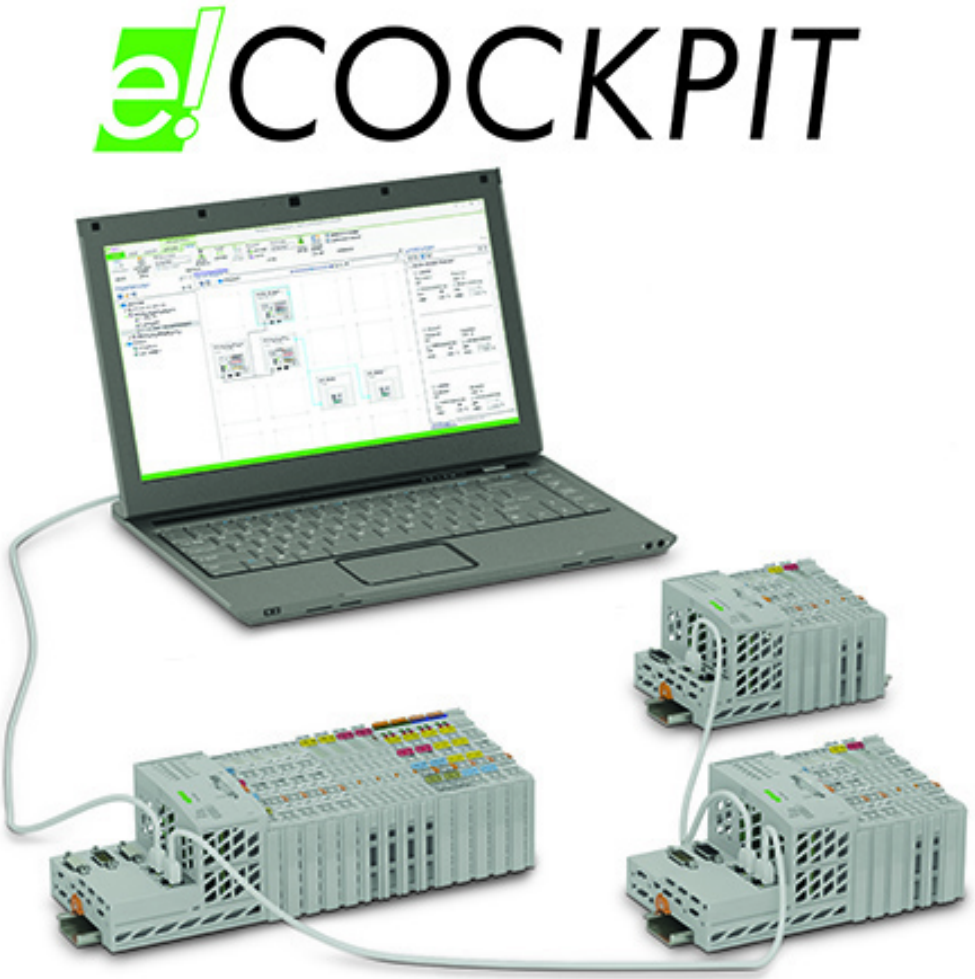
Documentation of the library WagoAppEtherNetIP_Adapter

Release 1.0.2.1



Contents

CHAPTER 1

Description

This document is automatically generated. Because of this, the chapter 30 Visualization is not shown in this document. If you are interested in getting to know more about visualization, we refer to the library manager of e!Cockpit.

Subject to Changes

WAGO Kontakttechnik GmbH Co. KG reserves the right to provide for any alterations or modifications. WAGO Kontakttechnik GmbH Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

Personnel Qualification

All tasks that are carried out with libraries made for the e!COCKPIT software must only be performed by qualified electrical specialists instructed in PLC programming according to IEC 61131-3.

All tasks that have an effect on the properties or the behavior of automation hardware or software products must only be performed by qualified employees with a thorough knowledge of handling the products concerned.

Intended Use of e!COCKPIT Libraries

Libraries created for the e!COCKPIT software are used to simplify the development of application projects in the IEC 61131-3 programming languages.

For automation tasks, WAGO offers programmable logic controllers in a wide variety of performance classes. In combination with a wide range of I/O modules, the controllers can process standard types of field signals. Controllers can be implemented centrally or in decentralized configurations. The controllers offer interfaces for the most commonly used fieldbuses for use in decentralized configurations. Fieldbus independent I/O modules are then linked via fieldbus couplers. WAGO controllers offer a runtime environment for user programs called e!RUNTIME. Software projects for implementation in e!RUNTIME environments can be created in e!COCKPIT. The programming environment in e!COCKPIT is based on the established CODESYS 3 industrial standard. Users with a previous knowledge of CODESYS 3 will thus find this environment largely familiar. The following programming languages of the IEC 61131-3 standard are available:

- Structured Text (ST)
- Ladder Diagram (LD)
- Function Block Diagram (FBD)
- Instruction List (IL)
- Sequential Function Chart (SFC)
- Continuous Function Chart (CFC)

The individual programming languages can also be combined as required during the development of the software. A portfolio of prepared libraries can be accessed for many frequently used functions in order to make software development more efficient. This document provides an overview of the WagoAppEtherNetIP_Adapter that WAGO offers for e!COCKPIT.

This library provides an EtherNet/IP Adapter implementation. [1]

Further library information are summerized here:

Company WAGO

Title WagoAppEtherNetIP_Adapter

Version 1.0.2.1

Categories WAGO BusinessView|Factory Automation; WAGO BusinessView|Process Automation; WAGO FunctionalView|Connectivity|FieldBus; WAGO LayerView|App; Application

Namespace WagoAppEtherNetIP_Adapter

Author WAGO / u013773

Placeholder WagoAppEtherNetIP_Adapter

[1]

Based on WagoAppEtherNetIP_Adapter.library, last modified 30.03.2020, 22:15:57.
The content of this file was automatically generated with None on 08.04.2020, 15:01:43

20 Program Organization Units

2.1 FbEtherNetIP_Adapter (FB)

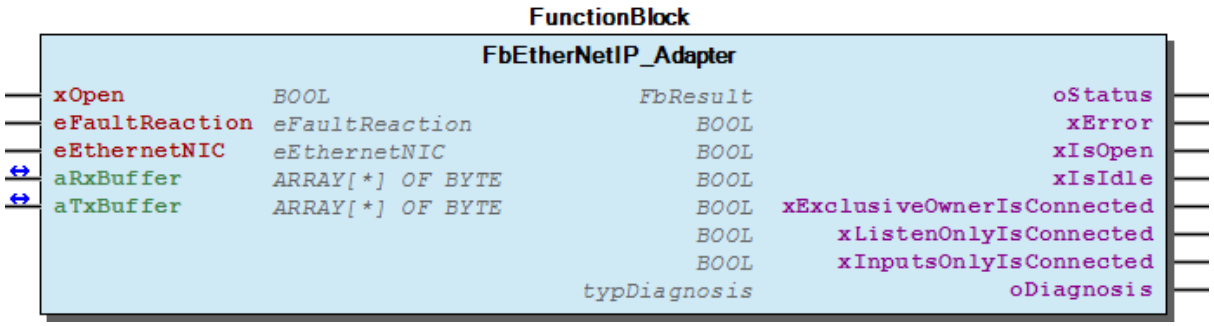
Interface variables

Scope	Name	Type	Comment
Input	xOpen	BOOL	Open communication channels of EtherNet/IP adapter
Output	oStatus	WagoSysError-Base.FbResult	Status information
	xError	BOOL	An Error has occurred. See oStatus for details
	xIsOpen	BOOL	EtherNet/IP adapter has opened its communication channels and is ready for scanner connections
	xIsIdle	BOOL	EtherNet/IP adapter is initialized and ready to be opened
Input	eFaultReaction	*eFaultReaction*	Defines the behaviour of aRxData in case of an interrupted fieldbus connection
	eEthernetNIC	*eEthernetNIC*	Determine Ethernet interface
Output	xExclusiveOwnerIsConnected	BOOL	Indicates status of Exclusive Owner connection: TRUE = Scanner has established a connection, FALSE = No scanner connected
	xListenOnlyIsConnected	BOOL	Indicates status of Listen Only connection: TRUE = Scanner has established a connection, FALSE = No scanner connected
	xInputsOnlyIsConnected	BOOL	Indicates status of Inputs Only connection: TRUE = Scanner has established a connection, FALSE = No scanner connected
	oDiagnosis	*typDiagnosis*	Diagnosis data
Input	aRxBuffer	POINTER TO BYTE	Byte array that holds the data which is received from the scanner (ORIGINATOR->TARGET)
	aTxBuffer	POINTER TO BYTE	Byte array that holds the data which is send to the scanner (TARGET->ORIGINATOR)

Function

This function block implements an EtherNet/IP Adapter.

Graphical Illustration



Dependencies

Note: Firmware Revision 02.05.23(08) or newer is requiered

Function Description

Note: This function block must be executed cyclically.

Note: Cycle time of e!COCKPIT task mus be at least as fast as the configured RPI. E.g. if the scanner uses a RPI of 10ms, the cycle time of e!COCKPIT task must be at least 10ms.

This function block implements an EtherNet/IP Adapter which allows up to three parallel class 1 connections (Exclusive Owner, Listen Only, Inputs Only) The maximum size of a connection is 500 Byte for each direction.

Behaviour:

Once the function block is called cyclically the output `xIsIdle` indicates that the adapter is ready to use. Otherwise `xError` indicates that the initialization has failed due to an error (e.g. incompatible firmware version).

If `xIsIdle` is `TRUE` and the input `xOpen` transits from `FALSE` to `TRUE` the function block will try to start the underlying EtherNet/IP stack. The output `xIsOpen` indicates that the start was successful and the function block can receive connection request from a scanner. If the underlying EtherNet/IP stack could not be started the output `xError` is set to `TRUE` and the output `oStatus` holds details about the reason.

When the input `xOpen` transits to `FALSE` again the function block will stop the underlying EtherNet/IP stack and all connections are disconnected.

The current connection state of the adapter is indicated by the outputs `xExclusiveOwnerIsConnected`, `xListenOnlyIsConnected` and `xInputsOnlyIsConnected`.

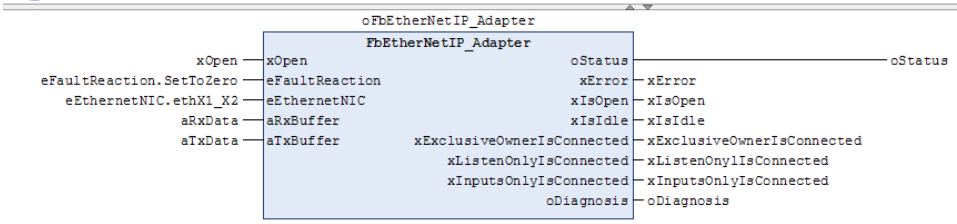
Example:

The connection size is determined by the size of the arrays `aRxBuffer` and `aTxBuffer`. The following declaration will setup a connection size of 32 Byte.

```
PROGRAM PLC_PRG
VAR
    oFbEtherNetIP_Adapter       : FbEtherNetIP_Adapter;
    xOpen                       : BOOL;
    oStatus                     : WagoAppEtherNetIP_Adapter.WagoSysErrorBase.FbResult;
    xError                      : BOOL;
    xIsOpen                     : BOOL;
    xIsIdle                     : BOOL;
    xExclusiveOwnerIsConnected  : BOOL;
    xListenOnylIsConnected      : BOOL;
    xInputsOnlyIsConnected      : BOOL;
    oDiagnosis                  : typDiagnosis;

    aRxData                     : ARRAY[1..32] OF BYTE; (* Array that holds the data which is received from the scanner (ORIGINATOR->TARGET) *)
    aTxData                     : ARRAY[1..32] OF BYTE; (* Array that holds the data which is send to the scanner (TARGET->ORIGINATOR) *)

    sTemp: STRING;
END_VAR
```



Class Overview

The following classes are accessible:

Identity Object (Class Code: 01 Hex)

Instance 1

ID	Access	Name	Data Type	Description	Default value
1	Get	Vendor ID	UINT	Manufacturer identification	40 (16#0028)
2	Get	Device Type	UINT	Indication of general type of product	12 (16#000C)
3	Get	Product Code	UINT	Identification of the particular product	2759 (16#0AC7)
4	Get	Revision	STRUCT of:	Revision	01 (16#01) 01 (16#00)
		Major Revision	*USINT*		
		Minor Revision	*USINT*		
5	Get	Status	WORD	Current status	0 (16#0000)
6	Get	Serial Number	UINT	Serial number	The last 4 digits of MAC ID
7	Get	Product Name	SHORT_STRING	Product name	WAGO 2759-101 EN/IP

Assembly Object (Class Code: 04 Hex)

Note: Writing of process data via explicit messaging is not supported!

Instance 100 (Output assembly)

ID	Access	Name	Data Type	Description	Default value
3	Get	Data	ARRAY of BYTE	Process data	

Instance 101 (Input assembly)

ID	Access	Name	Data Type	Description	Default value
3	Get	Data	ARRAY of BYTE	Process data	

Instance 102 (Dummy - needed for compatibility purposes)

ID	Access	Name	Data Type	Description	Default value
3	Get	Data	ARRAY of BYTE	Process data	

TCP/IP Interface Object (Class Code: F5 Hex)

Instance 0 (Class Attributes)

ID	Access	Name	Data Type	Description	Default value
1	Get	Revision	UINT	Revision of this object	4 (16#0004)

Instance 1

ID	Access	Name	Data Type	Description	Default value
1	Get	Status	DWORD	Interface status	1 (16#00 00 00 01)
2	Get	Configuration Capability	DWORD	Interface capability flags	32 (16#00 00 00 20)
3	Get	Configuration Control	DWORD	Interface control flags	0 (16#00 00 00 00)
4	Get	Physical Link Object	STRUCT of:	Path to physical link object	
		Path size	*UINT*	Size of Path	2 (16#00 02)
		Path	*Padded EPATH*	Logical segments identifying the physical link object	(16#20 F6 24 01)
5	Get	Interface Configuration	STRUCT of:	TCP/IP network interface config	
		IP Address	*UDINT*	IP address	Depends on TCP/IP config
		Network Mask	*UDINT*	Network mask	Depends on TCP/IP config
		Gateway Address	*UDINT*	Default gateway	Depends on TCP/IP config
		Name Server	*UDINT*	Primary name server	Depends on TCP/IP config
		Name Server 2	*UDINT*	Secondary name server	Depends on TCP/IP config
		Domain Name	*UDINT*	Default domain name	Depends on TCP/IP config
6	Get	Host Name	STRING	Host name	Depends on TCP/IP config
13	Get	Encapsulation Inactivity Timeout	UINT	Number of seconds of inactivity before TCP connection is closed	120

Ethernet Link Object (Class Code: F6 Hex)

Instance 0 (Class Attributes)

ID	Access	Name	Data Type	Description	Default value
1	Get	Revision	UINT	Revision of this object	4 (16#0004)

Instance 1

ID	Access	Name	Data Type	Description	Default value
1	Get	Interface Speed	UDINT	Interface Speed currently in use	Speed in Mbps(10,100, etc) Depends on link
2	Get	Interface Flags	DWORD	Interface status flags	
3	Get	Physical Address	ARRAY of 6 USINTs	MAC layer address	MAC ID of the interface the adapter is bind to
11	Get	Interface Capability	STRUCT of:	Indication of capabilities of the interface	
		Capability Bits	*DWORD*	Interface capabilities	Depends on TCP/IP config
		Speed/Duplex Options	*STRUCT of:*	Indicates speed/duplex pairs supported in the Interface Control attribute	
			USINT	Speed/Duplex Array Size	Depends on TCP/IP config
			ARRAY of STRUCT of:	Speed/Duplex Array	
			UINT	Interface Speed	Depends on TCP/IP config
			USINT	Interface duplex mode	Depends on TCP/IP config

29 Types

3.1 eEthernetNIC (ENUM)

InOut:

Name	Initial	Comment
ethX1_X2	0	Switched mode
ethX1	1	Separated mode X1
ethX2	2	Separated mode X2
br0	3	Use interface(s) assigned to br0
br1	4	Use interface(s) assigned to br1
br2	5	Use interface(s) assigned to br2
br3	6	Use interface(s) assigned to br3

3.2 eFaultReaction (ENUM)

InOut:

Name	Initial	Comment
SetToZero	0	Set the Rx data array to zero in case of communication loss
KeepLastValue	1	Hold the last values of Rx data array in case of communication loss

{attribute ‘qualified_only’}

3.3 typDiagnosis (STRUCT)

InOut:

Name	Type	Comment
uiInputDataSize	UINT	size of input data in bytes
uiOutputDataSize	UINT	size of output data in bytes
uiEstablished-Class1Connections	UINT	number of currently successfull established connections
udiConnectionTimeouts	UDINT	number of connection timeouts since the adapter started
udiForwardOpenRequests	UDINT	total number of forward open requests processed by the adapter
udiForwardOpenFails	UDINT	number of unsuccessfull forward open requests
uiInputOnlyConnected	UINT	How many input only connections are currently opened
uiListenOnlyConnected	UINT	How many listen only connections are currently opened
uiExclusiveOwnerConnected	UINT	How many exclusive owner connections are currently opened

80 Status

4.1 Status (GVL)

Scope	Name	Type
Constant	StatusEther-NetIP_Adapter	ARRAY [0..25] OF WagoTypesErrorBase.typResultItem

Description: Status Information

Value	Level	Description
ANY_TO_UINT(eStatus.OK	WagoTypesError-Base.WagoTypes.eSeverity.info	‘OK’
ANY_TO_UINT(eStatus.ERROR_INCOMPATIBLE_FIRMWARE	WagoTypesError-Base.WagoTypes.eSeverity.error	‘Firmware version is not compatible. At least 02.05.23(08) is needed ‘
ANY_TO_UINT(eStatus.ERROR_CHECK_FIRMWARE_FAILED	WagoTypesError-Base.WagoTypes.eSeverity.error	‘Check firmware has failed’
ANY_TO_UINT(eStatus.ERROR_CONFIG_EXBUFFER	WagoTypesError-Base.WagoTypes.eSeverity.error	‘Configuration of input data failed’
ANY_TO_UINT(eStatus.ERROR_CONFIG_EXBUFFER	WagoTypesError-Base.WagoTypes.eSeverity.error	‘Configuration of output data failed’
ANY_TO_UINT(eStatus.ERROR_CONFIG_ETHERNET	WagoTypesError-Base.WagoTypes.eSeverity.error	‘Configuration of Ethernet interface failed’
ANY_TO_UINT(eStatus.ERROR_CONFIG_VENDORINFO	WagoTypesError-Base.WagoTypes.eSeverity.error	‘Configuration of vendor specific information failed’
ANY_TO_UINT(eStatus.ERROR_START_COMMAND	WagoTypesError-Base.WagoTypes.eSeverity.error	‘Start of EtherNet/IP stack failed’
ANY_TO_UINT(eStatus.ERROR_STOP_COMMAND	WagoTypesError-Base.WagoTypes.eSeverity.error	‘Stop of EtherNet/IP stack failed’
ANY_TO_UINT(eStatus.ERROR_RESET_COMMAND	WagoTypesError-Base.WagoTypes.eSeverity.error	‘Reset of EtherNet/IP connections failed’
ANY_TO_UINT(eStatus.ERROR_START_COMMAND_TIMEOUT	WagoTypesError-Base.WagoTypes.eSeverity.error	‘Timeout while start of EtherNet/IP stack’
ANY_TO_UINT(eStatus.ERROR_OPEN_PORTS	WagoTypesError-Base.WagoTypes.eSeverity.error	‘EtherNet/IP adapter stack could not open Ethernet ports (Maybe already in use)’
ANY_TO_UINT(eStatus.ERROR_STACK	WagoTypesError-Base.WagoTypes.eSeverity.error	‘EtherNet/IP adapter stack returns error’
ANY_TO_UINT(eStatus.STATE_INIT	WagoTypesError-Base.WagoTypes.eSeverity.info	‘State: Init’
ANY_TO_UINT(eStatus.STATE_CHECK_FIRMWARE	WagoTypesError-Base.WagoTypes.eSeverity.info	‘State: Check firmware’
ANY_TO_UINT(eStatus.STATE_IDLE	WagoTypesError-Base.WagoTypes.eSeverity.info	‘State: Idle’
ANY_TO_UINT(eStatus.STATE_CONFIGURE	WagoTypesError-Base.WagoTypes.eSeverity.info	‘State: Configure’
ANY_TO_UINT(eStatus.STATE_START	WagoTypesError-Base.WagoTypes.eSeverity.info	‘State: Start’
ANY_TO_UINT(eStatus.STATE_STARTING	WagoTypesError-Base.WagoTypes.eSeverity.info	‘State: Starting’
ANY_TO_UINT(eStatus.STATE_RUN	WagoTypesError-Base.WagoTypes.eSeverity.info	‘State: Run’
ANY_TO_UINT(eStatus.STATE_STOP	WagoTypesError-Base.WagoTypes.eSeverity.info	‘State: Stop’
ANY_TO_UINT(eStatus.STATE_RESET	WagoTypesError-Base.WagoTypes.eSeverity.info	‘State: Reset’
ANY_TO_UINT(eStatus.STATE_ERROR	WagoTypesError-Base.WagoTypes.eSeverity.info	‘State: Error’
ANY_TO_UINT(eStatus.STATE_FATAL_ERROR	WagoTypesError-Base.WagoTypes.eSeverity.info	‘State: Fatal Error’

4.2 eStatus (ENUM)

InOut:

Name	Initial	Comment
OK	0	
ERROR_INCOMPATIBLE_FIRMWARE	EAPP + 1	Firmware version is not compatible. At least 02.05.23(08) is needed
ERROR_CHECK_FIRMWARE_FAILED	EAPP + 2	Firmware version could not be determined
ERROR_CONFIG_RXBUFFER	EAPP + 3	Configuration of Rx-Buffer failed
ERROR_CONFIG_TXBUFFER	EAPP + 4	Configuration of Tx-Buffer failed
ERROR_CONFIG_ETHERNET	EAPP + 5	Ethernet configuration failed
ERROR_CONFIG_VENDORINFO	EAPP + 6	Configuration of vendor specific information failed
ERROR_START_COMMAND	EAPP + 7	Execution of start command failed
ERROR_STOP_COMMAND	EAPP + 8	Execution of stop command failed
ERROR_RESET_COMMAND	EAPP + 9	Execution of reset command failed
ERROR_START_COMMAND_TIMEOUT	EAPP + 10	Transition to “Run” could not be performed within timeout
ERROR_OPEN_PORTS	EAPP + 11	EtherNet/IP adapter stack could not open Ethernet ports (Maybe already in use)
ERROR_STACK	EAPP + 12	EtherNet/IP adapter stack returns an error
STATE_INIT	EAPP + 100	Init
STATE_CHECK_FIRMWARE	EAPP + 101	Check firmware
STATE_IDLE	EAPP + 102	Idle
STATE_CONFIGURE	EAPP + 103	Configure
STATE_START	EAPP + 104	Start
STATE_STARTING	EAPP + 105	Starting
STATE_RUN	EAPP + 106	Run
STATE_STOP	EAPP + 107	Stop
STATE_RESET	EAPP + 108	Reset
STATE_ERROR	EAPP + 109	Error
STATE_FATAL_ERROR	EAPP + 110	Fatal Error

Description: Status Information

ParameterList (PARAMS)

InOut:

Scope	Name	Type	Initial	Comment
Constant	SCHEDULINGMODE	Wago-Types.eSchedulingMode	Wago-Types.eSchedulingMode.Background	Scheduling Mode for the background task that handles the Explicit Messaging

VersionHistory (GVL)

Name	Type
Info	ProjectInfo

WagoAppEtherNetIP_Adapter.library

date	version	author	change
30.01.2020	1.0.2.1	u013773	Add support for br2 and br3
25.01.2019	1.0.1.1	u013773	Update placeholder
08.01.2019	1.0.1.0	u015842	Properties: free placeholder added
28.10.2016	1.0.0.1	u013773	Update placeholder
04.10.2016	1.0.0.0	u013773	First release

Release Notes: Firmware version >= 02.05.23(08) for PFC100/PFC200 required Trigger mode “Change of State” is not supported for EtherNet/IP connections

CHAPTER 7

Library Reference

This is a dictionary of all referenced libraries and their name spaces.

Standard
Library Identification:
Placeholder: Standard
Default Resolution: Standard, * (System)
Namespace: Standard

Library Properties:

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

SysTask
Library Identification:
Placeholder: SysTask
Default Resolution: SysTask, * (System)
Namespace: SysTask

Library Properties:

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

WagoAppFileDir
Library Identification:
Placeholder: WagoAppFileDir
Default Resolution: WagoAppFileDir, * (WAGO)
Namespace: WagoAppFileDir

Library Properties:

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

Library Parameter:
Parameter: STANDARDSCHEDULINGMODE = eSchedulingMode.Background
Parameter: STANDARDTIMEOUT = TIME#10s0ms

WagoAppString
Library Identification:
Placeholder: WagoAppString
Default Resolution: WagoAppString, * (WAGO)
Namespace: WagoAppString

Library Properties:

- LinkAllContent: False
- Optional: False
- QualifiedOnly: False
- SystemLibrary: False
- PublishSymbolsInContainer: True

WagoSysAsync
Library Identification:
Placeholder: WagoSysAsync
Default Resolution: WagoSysAsync, * (WAGO)
Namespace: WagoSysAsync

Library Properties:

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

WagoSysBehaviourModels
Library Identification:
Placeholder: WagoSysBehaviourModels
Default Resolution: WagoSysBehaviourModels, * (WAGO)
Namespace: WagoSysBehaviourModels

Library Properties:

- LinkAllContent: False
- Optional: False
- QualifiedOnly: False

- SystemLibrary: False
- PublishSymbolsInContainer: True

WagoSysErrorBase
Library Identification:
Placeholder: WagoSysErrorBase
Default Resolution: WagoSysErrorBase, * (WAGO)
Namespace: WagoSysErrorBase

Library Properties:

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

Library Parameter:
Parameter: RES_LOG_MAX_FILESIZE = 2000
Parameter: RES_LOG_MAX_FILES = 1
Parameter: RES_LOG_MAX_ENTRIES = 200
Parameter: RES_LOG_NAME = 'WagoAppResultLogger'

WagoSysEtherNetIP_Adapter
Library Identification:
Placeholder: WagoSysEtherNetIP_Adapter
Default Resolution: WagoSysEtherNetIP_Adapter, * (WAGO)
Namespace: WagoSysEtherNetIP_Adapter

Library Properties:

- LinkAllContent: False
- QualifiedOnly: True
- SystemLibrary: False
- Optional: False

Library Parameter:
Parameter: BYCONFIGASSEMBLYID = 16#66
Parameter: BYCONSUMINGASSEMBLYID = 16#64
Parameter: BYDUMMYCONSUMINGASSEMBLY2ID = 16#C6
Parameter: BYPRODUCINGASSEMBLYID = 16#65
Parameter: BYDUMMYCONSUMINGASSEMBLY1ID = 16#C6

WagoSysVersion
Library Identification:
Name: WagoSysVersion
Version: 1.0.0.0
Company: WAGO

Namespace: WagoSysVersion

Library Properties:

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

WagoTypesCommon
Library Identification:
Placeholder: WagoTypesCommon
Default Resolution: WagoTypesCommon, * (WAGO)
Namespace: WagoTypes

Library Properties:

- LinkAllContent: False
- Optional: False
- QualifiedOnly: False
- SystemLibrary: False
- PublishSymbolsInContainer: True