

# Migration Notes

## EtherNet/IP Firmware V3

V3.8.0.3EN | Revision V3.8.0.3 | English | Released | Public | 2024-04-02



# Table of Contents

<b>1 Introduction .....</b>	<b>3</b>
1.1 About this document .....	3
1.2 References .....	3
1.3 Purpose and General Migration Aspects .....	3
<b>2 Migration Notes for the EtherNet/IP Firmware V3 .....</b>	<b>4</b>
2.1 Hotfix V3.8.0.2 .....	4
2.1.1 Reduce number of supported TCP, CL3 and UCCM connections from 8 to 7 in netX52-based targets .....	4
2.2 Release V3.8.0.0 .....	4
2.2.1 Finally remove the legacy support for HIL_SET_HANDSHAKE_CONFIG_REQ .....	4
2.2.2 Do not reflect the current Trigger Type in the Common Status Block .....	4
2.3 Release V3.7.0.0 .....	4
2.4 Release V3.6.0.0 .....	4
2.5 Release V3.5.0.0 .....	4
2.5.1 Support for HIL_SET_FW_PARAMETERS_REQ and HIL_GET_FW_PARAMETERS_REQ removed .....	4
2.5.2 Synchronization Handshake Mode changed .....	4
2.5.3 Substitution of HIL_SET_HANDSHAKE_CONFIG_REQ by HIL_SET_TRIGGER_TYPE_REQ .....	4
2.5.4 PD_IN handshake mode not reset to "free running" implicitly anymore .....	5
2.5.5 Removal of EIP_OBJECT_ID_SETDEVICEINFO_REQ .....	5
2.5.6 Synchronous EIP_OBJECT_CONNECTION_IND for opening of class 3 connections .....	5
2.5.7 Special behavior of SET_ATTRIBUTE_SINGLE of the Serial Number attribute of the Identity Object .....	5
2.5.8 Change of Remanent Data Handling .....	5
2.5.9 Harder tests for valid Assembly Flags .....	5
2.5.10 Changes in Assembly Flags and Attribute Option Flags .....	6
2.5.11 EIP_OBJECT_MR_REGISTER_REQ does not imply HIL_REGISTER_APP_REQ anymore .....	6
2.5.12 Support Set_Attribute_Single/Get_Attribute_Single for the Message Router Object .....	6
2.5.13 Support Set_Attribute_Single/Get_Attribute_Single for the Connection Manager Object .....	6
2.5.14 Changed status code of HIL_CHANNEL_INIT_REQ/CNF .....	6
2.5.15 EIP_APS_SET_CONFIGURATION_PARAMETERS_REQ does not delete remanent data implicitly anymore .....	6
2.5.16 Zero value for Product Code and Device Type in SET_CONFIGURATION_PARAMETERS_REQ no longer apply sophisticated default values .....	6
2.5.17 Attribute changes that are indicated via an Object Change Indication are now taken over implicitly by the protocol stack .....	7
2.5.18 Parameter Set V1 and V2 of EIP_APS_SET_CONFIGURATION_PAREMETERS_REQ are no longer supported .....	7
2.5.19 EIP_APS_SET_CONFIGURATION_REQ (0x3608) is not supported any longer .....	7
2.5.20 Reset handling .....	7
2.6 Hotfix V3.4.0.9 .....	7
2.6.1 Command EIP_OBJECT_SET_PARAMETER_REQ extended .....	7
2.7 Release V3.4.0.0 .....	7
2.7.1 Reset handling .....	7
2.7.2 Configurable PTP Pulse Width .....	8
2.7.3 New Handshake Mode: True interrupt driven cyclic data receive .....	8
2.7.4 CIP Identity object supports service "Flash_LEDs" (0x4b) now .....	8
2.7.5 PHYs are disabled until first configuration .....	8
2.7.6 Timeouts on indications .....	8
2.7.7 Synchronous Object Change Indications .....	8
2.8 Hotfix V3.3.0.11 .....	9
2.8.1 PHYs are disabled until first configuration .....	9
2.9 Release V3.3.0.0 .....	9
2.10 Release V3.2.0.0 .....	9
2.11 Release V3.1.0.0 .....	9



# Chapter 1 Introduction

## 1.1 About this document

This document briefly describes the changes between each two successive releases of this generation of the EtherNet/IP protocol firmware. For each new release the major differences compared to the previous release are documented. The intended audience is application software developers, testers and users. The range of topics is not limited, all user-observable changes may be described here, though this mostly should be about the packet API.

## 1.2 References

This document refers to the following documents:

- [1] Hilscher Gesellschaft fuer Systemautomation mbH: Protocol API, EtherNet/IP Adapter/Scanner V3.8.0.3.

## 1.3 Purpose and General Migration Aspects

This document lists general changes between different releases and provides a rough survey of the changes. The purpose of this document is to serve as an aid for porting of host applications towards each version boundary. This list may not be comprehensive, nor contains information on firmware-internal changes. It just outlines, on a higher level of abstraction, those modifications interfering with the behavior of the host application. For an overview of the particular stimuli of each modification and further descriptions refer to the Release Notes document and the JIRA ticket database. Please also take into account the Protocol API Manual [1] for more detailed descriptions on the mentioned topics. Migration between each two versions is additive, i.e. migration between two versions X and Z may require two (or even more) successive migration steps: from X to Y and from Y two Z.

This document focuses on the EtherNet/IP-specific functionality of our host application and aids us through the process of migrating it between different firmware versions. It does not cover those aspects related to services and mechanisms which are not specific to the EtherNet/IP firmwares, but the underlying operating system, the netX hardware configuration and bootstrapping, the middleware services, the firmware startup and update procedures, the number of communication channels and the functionality provided on each of these channels. There have been major changes between the V3 and the V5 major versions regarding these basic building blocks. Please take into account that these changes are not covered by this document. For EtherNet/IP, behavior is kept consistent between V3 and V5 version pairs as follows:

Compatible Version V3 Generation	Compatible Version V5 Generation
V3.5.0.x	V5.1.0.x
V3.6.0.x	V5.2.0.x
V3.7.0.x	V5.3.0.x
V3.8.0.x	V5.4.0.x

**NOTE** | The migration notes have to be read in conjunction with those of the stack core component used by the firmware version.

## Chapter 2 Migration Notes for the EtherNet/IP Firmware V3

### 2.1 Hotfix V3.8.0.2

#### 2.1.1 Reduce number of supported TCP, CL3 and UCCM connections from 8 to 7 in netX52-based targets

To address memory shortages in netX52-based EtherNet/IP firmwares, the number of supported parallel TCP server connections/sockets is reduced from 8 to 7. For technical reasons, this naturally also reduces the number of supported parallel UCCM requests and the number of concurrent class 3 connections similarly from 8 to 7.

This change has been introduced with ticket [PSEISV3-836](#).

### 2.2 Release V3.8.0.0

#### 2.2.1 Finally remove the legacy support for HIL\_SET\_HANDSHAKE\_CONFIG\_REQ

The command has been functionally substituted with HIL\_SET\_TRIGGER\_TYPE\_REQ some time ago, so that support for it can finally and entirely be removed from the firmware.

This change has been introduced with ticket [PSEISV3-804](#).

#### 2.2.2 Do not reflect the current Trigger Type in the Common Status Block

The trigger type can be set with HIL\_SET\_TRIGGER\_TYPE\_REQ and get with HIL\_GET\_TRIGGER\_TYPE\_REQ, providing a complete and consistent interface. Thus, the trigger type will no longer be observable through the common status block. The fields bPDInHskMode, bPDInSource, bPDOutHskMode, bPDOutSource, bSyncHskMode and bSyncSource will always be set to constant values as described.

This change has been introduced with ticket [PSEISV3-808](#).

### 2.3 Release V3.7.0.0

No Migration Notes have been written for this Version

### 2.4 Release V3.6.0.0

No Migration Notes have been written for this Version

### 2.5 Release V3.5.0.0

#### 2.5.1 Support for HIL\_SET\_FW\_PARAMETERS\_REQ and HIL\_GET\_FW\_PARAMETERS\_REQ removed

Support for the services HIL\_SET\_FW\_PARAMETERS\_REQ and HIL\_GET\_FW\_PARAMETERS\_REQ (which are aliases for RCX\_SET\_FW\_PARAMETERS\_REQ and RCX\_GET\_FW\_PARAMETERS\_REQ) is discontinued.

This change has been introduced with ticket [PSEISV3-507](#).

#### 2.5.2 Synchronization Handshake Mode changed

For convenience, the handshake mode for the synchronization handshake of the TimeSync object has been changed from RCX\_IO\_MODE\_BUFF\_DEV\_CTRL to RCX\_IO\_MODE\_BUFF\_HST\_CTRL. Semantically, this means that the handshake flags are one the host side in the very beginning and must initially be toggled to the netX for the synchronization trigger to be activated. Please carefully consider the changes in the example application "ExtendedConfigExample".

This change has been introduced with ticket [PSEISV3-425](#).

#### 2.5.3 Substitution of HIL\_SET\_HANDSHAKE\_CONFIG\_REQ by HIL\_SET\_TRIGGER\_TYPE\_REQ

The new packet HIL\_SET\_TRIGGER\_TYPE\_REQ was introduced as a substitute for the legacy packet HIL\_SET\_HANDSHAKE\_CONFIG\_REQ to improve comprehensibility. The handshake mode "Device controlled" can not be configured anymore if the new packet is used. For rcX-based firmwares the HIL\_SET\_HANDSHAKE\_CONFIG\_REQ command

will still be supported, but is marked as deprecated and its support may be removed entirely in future releases.

This change has been introduced with ticket [PSEISV3-425](#).

#### 2.5.4 PD\_IN handshake mode not reset to "free running" implicitly anymore

Previously, the PD\_IN handshake mode "RX triggered" was reset to the default mode "free running" during channel initialization. This behavior has been changed. The PD\_IN handshake mode, once set, will be kept until explicitly set to another mode by the application or until the netX device is rebooted.

This change has been introduced with ticket [PSEISV3-425](#).

#### 2.5.5 Removal of EIP\_OBJECT\_ID\_SETDEVICEINFO\_REQ

The support for command EIP\_OBJECT\_ID\_SETDEVICEINFO\_REQ is discontinued. Please use EIP\_OBJECT\_CIP\_SERVICE\_REQ to set the Device Identity attributes on a per-attribute basis.

This change has been introduced with ticket [PSEISV3-411](#).

#### 2.5.6 Synchronous EIP\_OBJECT\_CONNECTION\_IND for opening of class 3 connections

Previously, when the state of a CIP connection changed, an indication EIP\_OBJECT\_CONNECTION\_IND was generated to inform the host application about the connection change. Since such connections could be opened in rapid succession, this leads to flooding of the DPM mailbox and consequently, the loss of packets.

To overcome this issue, the new version of the EtherNet/IP stack waits for the host application to reply to the EIP\_OBJECT\_CONNECTION\_IND before the opening of a class 3 connection is completed. This assures that the maximum number of indications of type EIP\_OBJECT\_CONNECTION\_IND that can be in the round trip with the host application is limited by the maximum supported number of class 3 connections and thus prevents the flooding.

This change has been introduced with ticket [PSEISV3-409](#).

#### 2.5.7 Special behavior of SET\_ATTRIBUTE\_SINGLE of the Serial Number attribute of the Identity Object

When a packet EIP\_OBJECT\_CIP\_SERVICE\_REQ with command CIP\_CMD\_SET\_ATTR\_SINGLE is retrieved to set attribute 6 (Device Serial Number) of the Identity Object instance 1 to a value of zero, then the Serial Number which is stored in the Security Memory or Flash Device Label is applied instead. For all values different from zero, the retrieved value is applied. This is consistent with the behavior formerly implemented by the EIP\_OBJECT\_ID\_SETDEVICEINFO\_REQ packet.

This change has been introduced with ticket [PSEISV3-399](#).

#### 2.5.8 Change of Remanent Data Handling

Prior to V3.5.0.x the netX firmware has stored remanent data when using the basic configuration packet data set. In case the extended configuration packet set was used, the host application had to store/handle remanent data. Now, the responsibility for the remanent data is configurable via the taglist of the firmware.

In case the host application stores remanent data:

Remanent data handling is now realized via the new indication packet HIL\_STORE\_REMANENT\_DATA\_IND (0x2F8E) and the new request packet HIL\_SET\_REMANENT\_DATA\_REQ (0x2F8C). Please see section "Remanent Data Handling" in the API manual. This also affects the handling of the indication packet EIP\_OBJECT\_CIP\_OBJECT\_CHANGE\_IND (0x1AFA). Please read the adapted description of this packet in the API manual.

This change has been introduced with ticket [PSEISV3-397](#).

#### 2.5.9 Harder tests for valid Assembly Flags

Checks for valid combinations of assembly flags and further configuration parameters have been improved. Invalid values will be rejected.

Specifically, in previous versions, the parameter uInputAssFlags within the packet EIP\_APS\_SET\_CONFIGURATION\_PARAMETERS\_REQ allowed bit 0 (EIP\_AS\_TYPE\_INPUT) to be cleared ('0'). Now, this application has to explicitly set this bit. Otherwise the packet will be rejected.

This change has been introduced with ticket [PSEISV3-374](#).

### 2.5.10 Changes in Assembly Flags and Attribute Option Flags

Assembly Flags were renamed, backward compatibility is maintained. Source code can easily be adapted.

Attribute Option Flags were numerically changed. You are required to adapt and recompile your host application when you use them.

This change has been introduced with ticket [PSEISV3-374](#).

### 2.5.11 EIP\_OBJECT\_MR\_REGISTER\_REQ does not imply HIL\_REGISTER\_APP\_REQ anymore

In the V4 stack generation, when the host application registered an additional object class at the message router via packet `EIP_OBJECT_MR_REGISTER_REQ`, an implicit `HIL_REGISTER_APP_REQ` was also performed, leading to indication packets being generated toward the host application. This implication was removed in the V5 stack generation. Unless a host application explicitly issues `HIL_REGISTER_APP_REQ`, no indications will be generated at all and thus, requests to any registered additional object classes will be rejected with status code 5 (`CIP_GSR_BAD_CLASS_INSTANCE`).

This change has been introduced with ticket [PSEISV3-374](#).

### 2.5.12 Support Set\_Attribute\_Single/Get\_Attribute\_Single for the Message Router Object

The mentioned services are now supported at instance level of the message router object, although the instance does not implement any attributes which could be addressed. You have to modify your STC file for ODVA's Conformance Test tool accordingly.

This change has been introduced with ticket [PSEISV3-374](#).

### 2.5.13 Support Set\_Attribute\_Single/Get\_Attribute\_Single for the Connection Manager Object

The mentioned service is now supported at instance level of the connection manager object, although the instance does not implement any attributes which could be addressed. You have to modify your STC file for ODVA's Conformance Test tool accordingly.

This change has been introduced with ticket [PSEISV3-374](#).

### 2.5.14 Changed status code of HIL\_CHANNEL\_INIT\_REQ/CNF

The service `HIL_CHANNEL_INIT_REQ/CNF` will now return the error code `ERR_HIL_NOT_CONFIGURED` in the confirmation packet if during channel initialization, no configuration was applied and thus, the channel is in a reset state subsequent to the operation. Formerly, `SUCCESS_HIL_OK` was returned. The change allows `HIL_CHANNEL_INIT_REQ` to provide feedback over the actually performed operation, i.e. applying a stored configuration (when configuration is done via the basic packet set) or resetting the channel if no stored configuration is available (typically done when configuration is done via the extended packet set).

This change has been introduced with ticket [PSEISV3-374](#).

### 2.5.15 EIP\_APS\_SET\_CONFIGURATION\_PARAMETERS\_REQ does not delete remanent data implicitly anymore

In previous versions, the remanent data has been deleted by the firmware anytime the content of the packet `EIP_APS_SET_CONFIGURATION_PARAMETERS_REQ` has changed. This is not the case anymore.

If changes in this packet should take effect (e.g. changing IP address method from DHCP to static), the remanent data needs to be deleted by the application in advance by using the command `HIL_DELETE_CONFIG_REQ`.

This change has been introduced with ticket [PSEISV3-374](#).

### 2.5.16 Zero value for Product Code and Device Type in SET\_CONFIGURATION\_PARAMETERS\_REQ no longer apply sophisticated default values

Previously, if zero was specified for Product Code and Device Type, the parameters were set to default values depending on the used netX chip. This logic was removed. Now, a product code of zero will be rejected. A device type of zero will be accepted. Unless values will be set, fixed default values will apply, which are not depending on the used netX chip anymore.



This change has been introduced with ticket [PSEISV3-374](#).

### 2.5.17 Attribute changes that are indicated via an Object Change Indication are now taken over implicitly by the protocol stack

In former stack versions the host application had to actively take over new attribute values by issuing a set attribute single request to the stack after replying to the change indication. From V3.5 and V5.1 on, the protocol stack will implicitly take over the attribute value if the host application accepted the change by replying to the change indication with `u1Sta` set to 0.

Sending a set attribute single service must not be done anymore.

This change has been introduced with ticket [PSEISV3-367](#).

### 2.5.18 Parameter Set V1 and V2 of `EIP_APS_SET_CONFIGURATION_PARAMETERS_REQ` are no longer supported

The parameter sets V1 and V2 are not supported anymore. Host applications must use parameter set V3. Switching to a newer parameter set can be done without side effects by just filling the additional parameters with default values as described in the API manual.

This change has been introduced with ticket [PSEISV3-366](#).

### 2.5.19 `EIP_APS_SET_CONFIGURATION_REQ` (0x3608) is not supported any longer

The protocol stack no longer supports the request packet `EIP_APS_SET_CONFIGURATION_REQ` (0x3608). It will be answered with error code `ERR_HIL_UNKNOWN_COMMAND`.

Instead the packet `EIP_APS_SET_CONFIGURATION_PARAMETERS_REQ` (0x3612) has to be used.

This change has been introduced with ticket [PSEISV3-365](#).

## 2.5.20 Reset handling

The reset protocol between the stack and the host application was changed. With the new version, the host application is explicitly required to issue a `RCX_CHANNEL_INIT_REQ`, which was probably also done before, but was not mandatory yet. In order to ensure conformance to the EtherNet/IP specification, your host application has to comply to the reset protocol which is determined by the selected Configuration Packet Set. Please follow the instructions on reset behavior in the Protocol API manual carefully.

This change has been introduced with ticket [PSEISV3-294](#).

## 2.6 Hotfix V3.4.0.9

### 2.6.1 Command `EIP_OBJECT_SET_PARAMETER_REQ` extended

Starting with version V3.4.0.9 the following new parameter flag is available:

`EIP_OBJECT_PRM_DISABLE_FLASH_LEDS_SERVICE`

It can be used to disable the `Flash_LEDs` service of the CIP Identity object (class 0x01) which is enabled by default. This is particularly required for host application which control the MS/NS LEDs themselves, i.e. have them wired with the APP-CPU instead of the COM-CPU.

This change has been introduced with ticket [PSEISV3-445](#).

## 2.7 Release V3.4.0.0

### 2.7.1 Reset handling

The reset protocol between the stack and the host application was changed. With the new version, the host application is explicitly required to issue a `RCX_CHANNEL_INIT_REQ`, which was probably also done before, but was not mandatory yet. In order to ensure conformance to the EtherNet/IP specification, your host application has to comply to the reset protocol which is determined by the selected Configuration Packet Set. Please follow the instructions on reset behavior in the Protocol API manual carefully.

This change has been introduced with ticket [PSEISV3-294](#).

### 2.7.2 Configurable PTP Pulse Width

The Hilscher-specific attribute 300 “Sync Parameters” of the Time Sync object was extended by the pulse width of the sync trigger signal. Backwards compatibility is given.

This change has been introduced with ticket [PSEISV3-237](#).

### 2.7.3 New Handshake Mode: True interrupt driven cyclic data receive

The stack now features a new Handshake Mode “RX Triggered” which is beneficial for two reasons:

Low-latency reception of process data frames: Reception of new process data frames for selected input assemblies will instantly trigger the DPM handshake flags, rapidly presenting data updates to the host application.

True event-based operation: While the default input handshake mode “Free Running” naturally leads to a DPM polling approach, even when technically implemented with interrupts, the new mode will block any read operation on the DPM input area until new data is received for an eligible assembly. Thus, it reduces the host’s effort to have recent IO data available. The event will be buffered with depth one if the receive event occurs while the host is in control over the DPM input area. Please note that the RX Triggered Mode is not meant to provide bus-synchronous operation for the host application.

This change has been introduced with ticket [PSEISV3-236](#).

### 2.7.4 CIP Identity object supports service "Flash\_LEDs" (0x4b) now

The CIP Identity object (class code 0x01) supports the service “Flash\_LEDs” (0x4b) now as defined in the CIP Volume 1 specification. You have to modify your STC file for ODVA’s Conformance Test tool accordingly. Applications/Devices that do not have/use CIP LEDs can switch off this service by using the packet command EIP\_OBJECT\_SET\_PARAMETER\_REQ (available since version V3.4.0.10).

This change has been introduced with ticket [PSEISV3-212](#).

### 2.7.5 PHYs are disabled until first configuration

Before the stack is entirely configured for the first time, the PHYs are disabled and the Link LEDs of both the netX and the peer Ethernet ports will be off.

This change has been introduced with ticket [PSEISV3-209](#).

### 2.7.6 Timeouts on indications

Certain indications are now imposed with a timeout of three seconds. If the host application fails to reply to the indication within this period of time, then the stack will continue to process the causal service and eventually responds with an error status CIP\_GSR\_EMBEDDED\_SERVICE\_ERROR (0x1E) on the network. You are encouraged to process all indications rapidly. Section Application Compliance in the Protocol API manual outlines these new requirements.

This change has been introduced with ticket [PSEISV3-190](#).

### 2.7.7 Synchronous Object Change Indications

Object change indications are generated by the EtherNet/IP stack towards the application for a subset of the implemented object’s attributes on value changes caused by network access.

The modifiable Attribute Options of each attribute decide over whether an indication is generated for a particular attribute to notify about each of its changes. Basically, all attributes which need -according to the specification - non-volatile storage are notified by default.

With the new version of the stack, attribute changes due to requests over the EtherNet/IP network are no longer applied directly, but are presented to the host application before they are applied onto the Object Dictionary. Thus, the host application is given the possibility to reject any unwanted modification, e.g. to further limit allowed value ranges of built-in attributes. Eventually, the host application still will have to actively apply the new attribute value presented with the Object Change Indication by re-issuing a corresponding service request over the DPM packet interface.





This change has been introduced with ticket [PSEISV3-190](#).

## 2.8 Hotfix V3.3.0.11

### 2.8.1 PHYs are disabled until first configuration

Before the stack is entirely configured for the first time, the PHYs are disabled and the Link LEDs of both the netX and the peer Ethernet ports will be off.

This change has been introduced with ticket [PSEISV3-209](#).

## 2.9 Release V3.3.0.0

No Migration Notes have been written for this Version

## 2.10 Release V3.2.0.0

No Migration Notes have been written for this Version

## 2.11 Release V3.1.0.0

No Migration Notes have been written for this Version