Technical Document

Niagara KNXnet/IP Driver Guide

October 22, 2019

Niagara KNXnet/IP Driver Guide

Tridium, Inc.
3951 Westerre Parkway, Suite 350
Richmond, Virginia 23233
U.S.A.

Confidentiality

The information contained in this document is confidential information of Tridium, Inc., a Delaware corporation ("Tridium"). Such information and the software described herein, is furnished under a license agreement and may be used only in accordance with that agreement.

The information contained in this document is provided solely for use by Tridium employees, licensees, and system owners; and, except as permitted under the below copyright notice, is not to be released to, or reproduced for, anyone else.

While every effort has been made to assure the accuracy of this document, Tridium is not responsible for damages of any kind, including without limitation consequential damages, arising from the application of the information contained herein. Information and specifications published here are current as of the date of this publication and are subject to change without notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia.

Trademark notice

KNX® and the KNX logo are registered trademarks of the KNX Association cvba. BACnet and ASHRAE are registered trademarks of American Society of Heating, Refrigerating and Air-Conditioning Engineers. Microsoft, Excel, Internet Explorer, Windows, Windows Vista, Windows Server, and SQL Server are registered trademarks of Microsoft Corporation. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Mozilla and Firefox are trademarks of the Mozilla Foundation. Echelon, LON, LonMark, LonTalk, and LonWorks are registered trademarks of Echelon Corporation. Tridium, JACE, Niagara Framework, NiagaraAX Framework, and Sedona Framework are registered trademarks, and Workbench, WorkPlaceAX, and AXSupervisor, are trademarks of Tridium Inc. All other product names and services mentioned in this publication that is known to be trademarks, registered trademarks, or service marks are the property of their respective owners.

Copyright and patent notice

This document may be copied by parties who are authorized to distribute Tridium products in connection with distribution of those products, subject to the contracts that authorize such distribution. It may not otherwise, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Tridium, Inc.

Copyright © 2019 Tridium, Inc. All rights reserved.

The product(s) described herein may be covered by one or more U.S. or foreign patents of Tridium.

About this guide

This topic contains important information about the purpose, content, context, and intended audience for this document.

Product Documentation

This document is part of the Niagara technical documentation library. Released versions of Niagara software include a complete collection of technical information that is provided in both online help and PDF format. The information in this document is written primarily for Systems Integrators. In order to make the most of the information in this book, readers should have some training or previous experience with Niagara 4 software, as well as experience working with JACE network controllers.

Document Content

This document describes how to set up and use the Knxnet/IP driver.

CAUTION: Protect against unauthorized access by restricting physical access to the computers and devices that manage your building model. Set up user authentication with strong passwords, and secure components by controlling permissions. Failure to observe these recommended precautions could expose your network systems to unauthorized access and tampering.

Document change log

Updates (changes and additions) to this document are listed below.

October 22, 2019

- Updated during the release of Niagara 4.8.
- Edited all topics for grammar and style.
- Replaced Properties chapter with Components chapter.
- Added screen captures and how to access properties.
- Reorganized FAQs.
- Created Troubleshooting topic.
- In the topic, “About this guide”, added a caution note alerting customers to restrict access to all computers, devices, field buses, components, etc., that manage their building model.

July 24, 2018

Updated for the 1.6 release of the driver.

2018

Editorial changes only.

January 11, 2017

Added FAQ’s and property details.

September 28, 2016

Added property details.

August 4, 2016

Added property topics.

July 26, 2016

- Added Property reference chapter and draft property topics.
- Edited About KNX DataDefs.

- Edited Points in Fault — KNXnet/IP Interface FAQ.
- Added topics for Add Devices to the KNX Network and Add Points.
- Added About, Examining and Updating KNX Data Defs.
- Added KNXnet/IP driver conflict FAQ.

April 04, 2016

Related documentation

Several other documents are available for learning how to use the Niagara Knxnet/IP driver.

- *Niagara Drivers Guide* explains concepts.
- *Getting Started with Niagara* explains concepts.

Contents

Chapter 1 About the KNXnet/IP driver

Topics covered in this chapter

- ♦ Compatibility
- ♦ Requirements
- ♦ Driver modules
- ♦ ETS FAQs
- ♦ Driver FAQs
- ♦ Tunnelling FAQs
- ♦ Compound Structures
- ♦ Quick start

The KNX decentralized, distributed network protocol manages building controls, such as HVAC and lighting systems. The Niagara Framework® KNX driver models devices that conform to the KNX protocol standard.

The KNXnet/IP driver:

- Provides a license feature: knxnetIp
- Works with NiagaraAX-3.8 and Niagara 4.1 versions and later
- Supports the JACE-3E, JACE-6, JACE-6E and JACE-8000 (it does not support the JACE-2)
- Supports both ETS4 (Engineering Tools Software) and ETS5 KNX tools
- Supports versions 12-14 of the ETS knxproject xml schema
- Uses the more data-rich ETS project file as a source of KNX system data for discovery
- Supports ETS Two Level, Three Level and Free Level group addressing
- Supports the full list of current KNX datapoint types defined in the knx Master Data xml file
- Supports manufacturer-specific KNX datapoint types with values exposed as hex strings
- Is ready for knxnetIp Secure and KNX Data Secure (Secure Application Layer)
- Supports enhanced point discovery to improve pre-setting Point Facets, Point Names, Hierarchy and Point types
- Supports complex multi-value KNX datapoint types

  Consistent modelling (knxnetIp-device = Niagara-device and KNX Group Address = Niagara-point)
- Supports knxnetIp tunnelling
- Supports proxy routing
- Supports enhanced bus-data-received functionality
- Provides documentation as a PDF and Workbench help
- Makes no reference to the EIB (European Installation Bus)

Compatibility

The `knxnetIp` driver is compatible with standard KNX Services.

KnxnetIp services compatibility

KnxnetIp Service	Support
Core Services	Fully supported
Device Management	Client only
Tunnelling	Data Link Layer — Fully Supported CEMI Raw— Not Supported KNX Busmonitor— Not Supported
Routing Service	Not Supported
Remote Logging	Not supported
Remote Configuration and Diagnostics	Not supported
Object Server	Not supported

Routing

Through a tunnel connection the driver supports physical routers, which route KNX messages.

The driver does not support the KNX protocol's Routing Service, which broadcasts messages around the entire network.

The driver provides proxy routing, which uses a tunnel to connect to a router device for the purpose of sending messages.

KNX system specification editions

Here are the specific KNX System Specification document editions:

- 03_01_02 Glossary v01.04.00 AS.pdf
- 03_02_06 Communication Medium KNX IP v01.00.02 AS.pdf
- 03_03_02 Data Link Layer General v01.02.02 AS.pdf
- 03_03_07 Application Layer v01.06.04 AS.pdf
- 03_06_03 EMI_IMI v01.03.03 AS.pdf
- 03_07_02 Datapoint Types v01.08.03 AS.pdf
- 03_07_03 Standardized Identifier Tables v01.03.01 AS.pdf
- 03_08_01 Overview v01.04.02 AS.pdf
- 03_08_02 Core v01.05.02 AS.pdf
- 03_08_03 Management v01.06.02 AS.pdf
- 03_08_04 Tunnelling v01.05.04 AS.pdf

Requirements

Requirements include installer training, ETS data, software, hardware, and licensing.

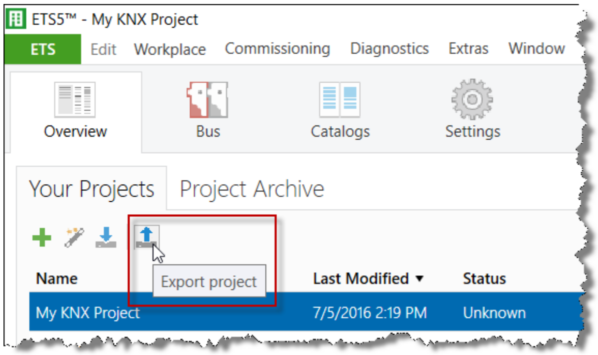
Systems integrator training

The procedures in this document assume that you are Niagara certified and experienced at configuring stations.

ETS project requirements

Using ETS (Engineering Tool Software), export the ETS Project. This provides the source of the KNX system data for the Knxnet/IP driver.

Figure 1 ETS Project view



Platform prerequisites

The KNXnet/IP driver requires a NiagaraAX or Niagara 4 platform that supports Hot Spot VM (virtual machine) from Oracle. It does not support the JACE 2, 4, or 5. The KNXnet/IP driver only supports an IP connection to the KNX network.

Version

These Niagara versions support the KNXnet/IP driver:

- NiagaraAX-3.8
- Niagara 4.1 and later

Licensing requirement

- The knxnetIp feature must be present in your Workbench platform and station platform licenses.
- Attributes associated with the knxnetIp feature are listed in the table.

Attribute	Description
device.limit	This attribute is common to most features. It defines the maximum number of devices that can be connected to this driver.
history.limit	This attribute is common to most features. It defines the maximum number of histories that can be used for this feature.
installation.limit	This attribute defines the maximum number of KNX installations that can be connected to this driver.
interface.limit	This attribute defines the maximum number of platform interfaces that can be connected to this driver.
point.limit	This attribute is common to most features. It defines the maximum number of points used on this feature.
schedule.limit	This attribute is common to most features. It defines the maximum number of schedules used on this feature.

Driver modules

KNX integration requires modules for both NiagaraAX and Niagara 4 versions.

NiagaraAX KNX modules

Two modules are required for the KNXnet/IP driver:

- knxnetIp.jar
- docknxnetIp.jar

Niagara 4 KNX modules

Three modules are required for the KNXnet/IP driver:

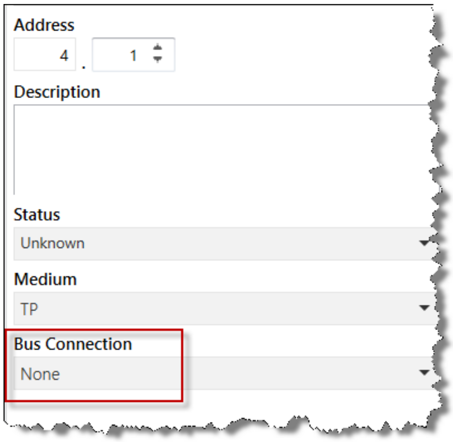
- knxnetIp-doc.jar
- knxnetIp-rt.jar
- knxnetIp-wb.jar

ETS FAQs

The implementation of a KNX network begins with the ETS project.

Does the 'Bus Connection' property require configuration in ETS?

Figure 2 Bus Connection property in ETS

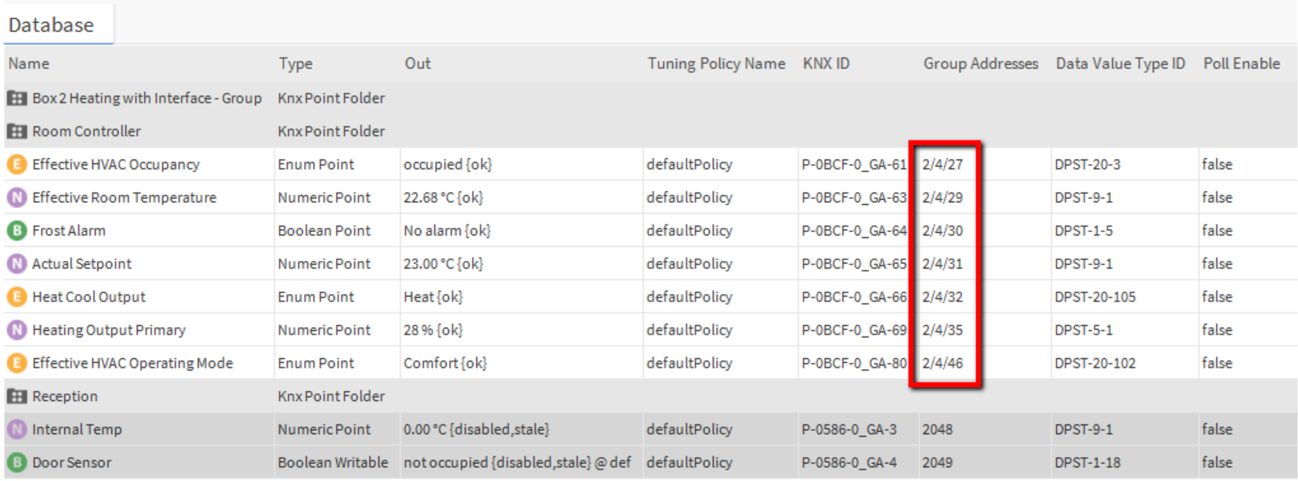


Address
4 . 1
Description
Status
Unknown
Medium
TP
Bus Connection
None

No. To import devices from an exported ETS Project file it is not necessary to configure the **Bus Connection** property in ETS.

Can I discover points from different ETS projects and add them to the same device?

Figure 3 Multi-level project



Database

Name	Type	Out	Tuning Policy Name	KNX ID	Group Addresses	Data Value Type ID	Poll Enable
Box 2 Heating with Interface - Group	Knx Point Folder						
Room Controller	Knx Point Folder						
Effective HVAC Occupancy	Enum Point	occupied {ok}	defaultPolicy	P-0BCF-0_GA-61	2/4/27	DPST-20-3	false
Effective Room Temperature	Numeric Point	22.68 °C {ok}	defaultPolicy	P-0BCF-0_GA-63	2/4/29	DPST-9-1	false
Frost Alarm	Boolean Point	No alarm {ok}	defaultPolicy	P-0BCF-0_GA-64	2/4/30	DPST-1-5	false
Actual Setpoint	Numeric Point	23.00 °C {ok}	defaultPolicy	P-0BCF-0_GA-65	2/4/31	DPST-9-1	false
Heat Cool Output	Enum Point	Heat {ok}	defaultPolicy	P-0BCF-0_GA-66	2/4/32	DPST-20-105	false
Heating Output Primary	Numeric Point	28 % {ok}	defaultPolicy	P-0BCF-0_GA-69	2/4/35	DPST-5-1	false
Effective HVAC Operating Mode	Enum Point	Comfort {ok}	defaultPolicy	P-0BCF-0_GA-80	2/4/46	DPST-20-102	false
Reception	Knx Point Folder						
Internal Temp	Numeric Point	0.00 °C {disabled,stale}	defaultPolicy	P-0586-0_GA-3	2048	DPST-9-1	false
Door Sensor	Boolean Writable	not occupied {disabled,stale} @ def	defaultPolicy	P-0586-0_GA-4	2049	DPST-1-18	false

Yes you can. You can also add points from different projects that use different Group Address Style strategies. It is possible to mix points from Two Level, Three Level and Free Group Address styles in the same device.

A password-protected dialogue opens when I try to discover devices and points. Why is this and what password do I enter?

The KNX ETS can export a project with a password. The KNXnet/IP driver can open the project when you present the correct credentials. You will need to enter the same password used to set up the ETS project.

Driver FAQs

What is a KNXnet/IP Interface device?

A KNXnet/IP Interface device is a hardware device that supports KNXnet/IP tunnelling only. A single interface, such as the Weinzierl 730 and the Siemens N 148/22 may support multiple KNXnet/IP tunnelling connections. These devices can have simultaneous tunnelling connections, which are managed by defining multiple KNX individual device addresses on the device. In the KNXnet/IP driver, this is called the Individual Device Address and allows the KNX network to be accessed by both ETS and the KNXnet/IP driver.

What is a KNXnet/IP router device?

This is a hardware device that supports both KNXnet/IP tunnelling and KNXnet/IP routing connections, as well as having the filter table to allow the device to perform as a coupler. Some KNXnet/IP Routers, such as the Siemens N 146/02 allow multiple tunnelling connections.

What is KNXnet/IP routing?

KNXnet/IP Routing is a multicast-based telegram, which allows a KNXnet/IP router to perform the function of a line or area coupler. This means the backbone of a KNXsystem can be Ethernet-based, allowing a much higher speed of transmission and more flexibility when installing. The KNXnet/IP router also provides a filter table to manage the flow of traffic where needed.

The KNXnet/IP driver does not support KNXnet/IP Routing.

What is proxy routing?

The KNXnet/IP driver does not directly support the KNXnet/IP Routing Service. Proxy routing:

- Enables the KNXnet/IP driver to communicate with KNXnet/IP router devices whose IP subnet differs from the Host IP subnet.
- Allows the KNXnet/IP driver to communicate with multiple KNXnet/IP router devices without using up a KNXnet/IP tunnelling connection in each KNXnet/IP router.

Proxy routing relies on configuring a KNXnet/IP device in the station using a KNXnet/IP tunnelling connection to one KNXnet/IP router device that has the same IP subnet address as the host. The KNXnet/IP router device's filtering needs to be configured to allow it to route messages from the KNXnet/IP tunnelling connection to other KNXnet/IP router devices and vice-versa.

The other KNXnet/IP routerdevices can then be configured as KNXnet/IP devices in the station with their **Connection Method** property set to `Proxy Routing` and their proxy device address and other connection properties set to the KNXnet/IP device that was configured to use the KNXnet/IP tunnelling connection described above.

What KNXnet/IP device address information does the KNX driver need and how does it obtain it?

Interfaces and routers are the two primary KNXnet/IP device types. The KNXnet/IP driver needs three addresses to communicate with either of these device types:

- **Ip Address**: Obtained from the KNXnet/IP device either over the KNX network using the driver's search-network-for-devices feature or by manual data entry.

- **Control Port Number**: Obtained from the KNXnet/IP device either over the KNX network using the driver's search-network-for-devices feature, by manual data entry, or by default.

  NOTE: The default **Control Port Number** is 3671.

- **Individual Device Address**: Obtained from the KNXnet/IP device either over the KNX network using the driver's search-network-for-devices feature, by manual data entry, or by using importing devices from the driver's`knxproj` file.

  NOTE: Manual data entry is not available in v29 of the KNXnet/IP driver.

The KNXnet/IP driver's search-network-for-devices (device discovery) feature uses a multicast connection.

NOTE: If both the **Ip Address** and **Control Port Number** are known by the KNXnet/IP driver, it does not initiate a search request via a multicast connection.

Does the KNXnet/IP driver support multicast telegrams?

The term "multicast" can apply to both IP Multicast and the KNX point-to-multi-point, connection-less (multicast) transport layer communication mode.

To facilitate device discovery on the network, the KNXnet/IP driver does support IP Multicast.

NOTE: The default **IP Multicast Address** is `224.0.23.12`.

The KNXnet/IP driver inherently supports the KNX point-to-multi-point, connection-less (multicast) transport layer communication mode in so far as it makes use of the A_GroupValue_Read-PDU, A_GroupValue_Response-PDU and A_GroupValue_Write-PDU Application Layer services, which are only supported using this communication mode.

Does the KNX driver support KNXnet/IP devices that do not support multicast?

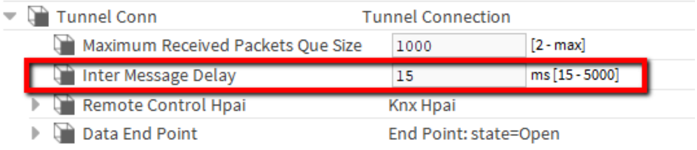
Yes, providing the **Ip Address** and **Control Port Number** are known to the KNXnet/IP driver. They can both be entered manually.

NOTE: Multicast–connection support is part of the Core: KNXnet/IP service, which is mandatory for all certified KNXnet/IP devices.

What controls the rate of KNX data requests from the driver?

The **Inter Message Delay** setting in the tunnel connection paces the out-going data requests from the driver. This can be used to reduce the rate of data requests in cases where a KNXnet/IP Interface device cannot cope with the traffic volume, caused possibly by its implementation settings or by its operating speed. The **Inter Message Delay** default setting of minimum 15ms has proved successful with Siemens interface devices. Any problems arising from this being set too small would manifest as control points intermittently having a Read Fault: `Timed out waiting for L_Data_con` (condition).

Figure 4 Inter Message Delay



Tunnel Conn	Tunnel Connection	
Maximum Received Packets Que Size	1000	[2 - max]
Inter Message Delay	15	ms [15 - 5000]
Remote Control Hpai	Knx Hpai	
Data End Point	End Point: state=Open	

Each out-going data request actually involves six packets travelling between the KNXnet/IP driver and the KNXnet/IP Interface as follows:

1. Request from KNXnet/IP driver to KNXnet/IP Interface
2. Acknowledgement from KNXnet/IP Interface to KNXnet/IP driver
3. Confirmation from KNXnet/IP Interface to KNXnet/IP driver
4. Acknowledgement from KNXnet/IP driver to KNXnet/IP Interface
5. Reply from KNXnet/IP Interface to KNXnet/IP driver

6. Acknowledgement from KNXnet/IP driver to KNXnet/IP Interface

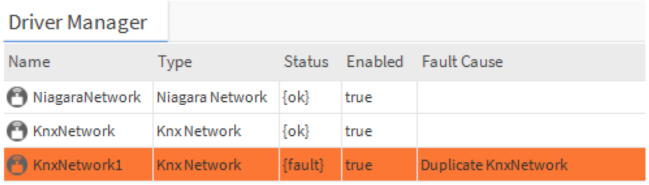
This appears as several flashes on the KNXnet/IP Interface's LEDs.

NOTE: There is an important difference in the implementation of **Inter Message Delay** between the KNXnet/IP driver and the EIBnet/IP driver. In the EIBnet/IP driver the **Inter Message Delay** was applied between all out-going messages, including acknowledgements and all connection control messages. The KNXnet/IP driver only applies the **Inter Message Delay** between data request messages.

Can I have more than one KNX network in my station?

No. There can only be one **KnxNetwork** in a station.

Figure 5 Driver Manager view



Driver Manager

Name	Type	Status	Enabled	Fault Cause
NiagaraNetwork	Niagara Network	{ok}	true	
KnxNetwork	Knx Network	{ok}	true	
KnxNetwork1	Knx Network	{fault}	true	Duplicate KnxNetwork

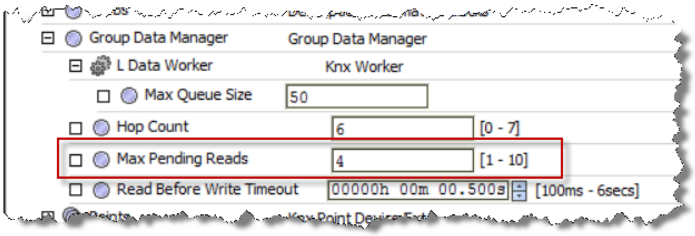
How does the driver prevent communications traffic from swamping the KNX twisted pair line?

The KNXnet/IP driver prevents the KNX twisted pair line from being swamped with traffic by controlling the number of concurrently active group address read operations and is set by the **Max Pending Reads** property in the **Group Data Manager**. The term "active" in this context means that:

1. A particular group address read operation reached the head of the group data operation queue.
2. The communications stack sent an L_Data_req message to the KNXnet/IP device and received an L_Data_con reply, but has not yet received a corresponding L_Data_ind message.

The default value of **Max Pending Reads** is 4 but, unfortunately, there is no clear guidance in the KNX Specs as to what an acceptable **Max Pending Reads** value should be. However, having this value too small does not cause a read–queue–is–full fault.

Figure 6 Max Pending Reads



Group Data Manager	Group Data Manager	
L Data Worker	Knx Worker	
Max Queue Size	50	
Hop Count	6	[0 - 7]
Max Pending Reads	4	[1 - 10]
Read Before Write Timeout	00000h 00m 00.500s	[100ms - 6secs]

Is the IP port adjustable?

This question is sometimes asked by IT departments who need control over the IP port usage on their network.

Protocol: The KNX System specifications specify that UDP (User Datagram Protocol) is used for KNXnet/IP connections rather than TCP (Transmission Control Protocol) and, therefore, the following commentary on ports and messaging is in relation to UDP.

Device and Driver Ports: There are four logical UDP ports allocated for each tunnel connection between the KNXnet/IP driver and a KNXnet/IP Interface device. Two ports (one for control and one for data) are at the device and two ports are at the driver (control and data). Activities, such as opening, closing, the connection and for heart beat use the control port. Point data use the data port.

Port numbers: It is usual that KNXnet/IP Interface devices use port 3671 for both control and data, but this may not always be the case. The KNXnet/IP driver dynamically allocates the two ports it uses to communicate with the device when establishing and maintaining a connection. In this way, the driver can separate the messaging between multiple devices concurrently by allocating different port numbers. In the driver, the control and data ports for each device are always different.

Port number selection: The ETS tool sets the port numbers used by the KNXnet/IP Interface device (provided that the device can be configured). The port numbers that the KNXnet/IP driver chooses dynamically for each Local Interface range between a port-minimum (0-65535) and port-maximum (0-65535). The default values of these are 3500 and 4000 although they can be changed when setting up of the driver. Clearly, reducing the range restricts the number of potential device connections and, ultimately, if the range offers only one port number, the driver will not function because two ports is a minimum for each device connection.

Recycling port numbers: When dynamically choosing a new port number, the KNXnet/IP driver starts at the port minimum number (default setting) and chooses the next available port number or it cycles through the range. You can select this behaviour when setting up the driver.

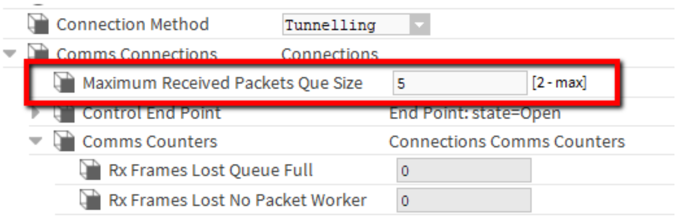
Tunnelling FAQs

KNXnet/IP tunnelling is the primary method of interfacing to a KNX system. It allows for Unicast communication from a single external device to the KNX system. This is akin to using a USB or Serial Interface to interface to the KNX system.

What maintains the tunnel connection?

The **Control End Point** connects, maintains and disconnects the tunnel connection. To be more precise, control of the tunnel connection is the only thing the **Control End Point** does. The **Control End Point** sees very little traffic, only two or three messages to connect or disconnect and one or two messages per minute to maintain the connection.

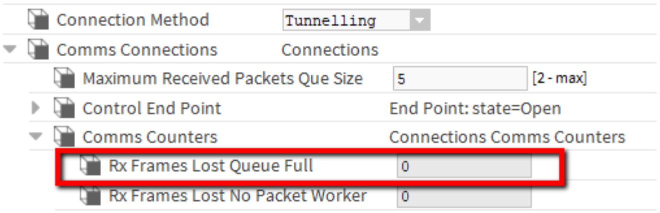
Figure 7 Maximum Received Packets Que Size



Connection Method	Tunnelling	
Comms Connections	Connections	
Maximum Received Packets Que Size	5	[2 - max]
Control End Point	End Point: state=Open	
Comms Counters	Connections Comms Counters	
Rx Frames Lost Queue Full	0	
Rx Frames Lost No Packet Worker	0	

The **Maximum Received Packets Que Size** controls the size of the control end point's receive queue. It defaults two five packets. If this value were too small, there would be an indication of lost frames in the hidden **Rx Frames Lost Queue Full** communications counter.

Figure 8 Rx Frames Lost Queue Full

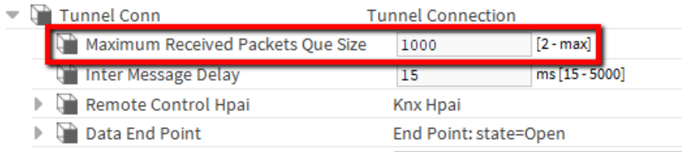


Connection Method	Tunnelling	
Comms Connections	Connections	
Maximum Received Packets Que Size	5	[2 - max]
Control End Point	End Point: state=Open	
Comms Counters	Connections Comms Counters	
Rx Frames Lost Queue Full	0	
Rx Frames Lost No Packet Worker	0	

Which queue is used for data through the tunnel connection?

The **Maximum Received Packets Que Size** controls the size of the tunnel connection's **Data End Point** receive queue. All **Group Address** messages pass through this queue and End Point so it is much busier than the **Control End Point**. Because it handles more traffic, its default queue size of 50 is larger than that of the **Control End Point**'s queue size.

Figure 9 Maximum Received Packets Que Size

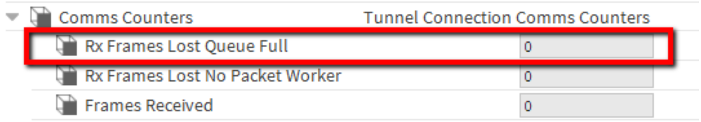


Tunnel Conn	Tunnel Connection	
Maximum Received Packets Que Size	1000	[2 - max]
Inter Message Delay	15	ms [15 - 5000]
Remote Control Hpai	Knx Hpai	
Data End Point	End Point: state=Open	

Unlike the **Group Data Operation Queue**, the necessary size of this queue does not depend on the number and frequency of control point polling. Instead, it depends on how quickly the KNXnet/IP driver can process incoming messages, which in turn depends on the overall CPU usage in the platform.

You can monitor the success or otherwise of received data by inspecting the **Rx Frames Lost Queue Full** counter in the hidden communications counters. Depending on the number and frequency of lost Rx frames you can try increasing the **Maximum Received Packets Que Size** in steps of 50, until no more Rx frames are being lost.

Figure 10 Rx Frames Lost Queue Full



Comms Counters	Tunnel Connection Comms Counters
Rx Frames Lost Queue Full	0
Rx Frames Lost No Packet Worker	0
Frames Received	0

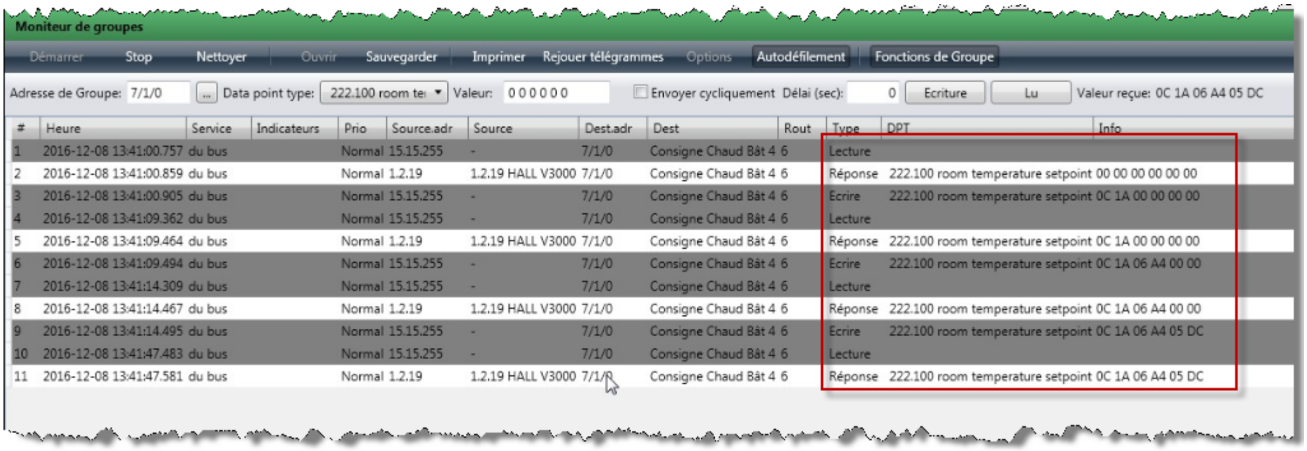
Compound Structures

The KNX standard specifies that some Datapoint Types (DPT) have compound structures, where one DTP contains several data fields. So, how does the KNXnet/IP driver write to a KNX compound structure DPT? This topic answers this question with an example.

Datapoint Type 222.100 is a DPT with three sets of room temperature setpoints, each of which is a 16-bit float value. The room temperature setpoint comfort (TempSetpComf), room temperature setpoint standby (TempSetpStdby) and room temperature setpoint economy (TempSetpEco) are all encoded within the same DPT.

From the standpoint of the framework's driver data model, each of these setpoints is a separate proxy point but, on the wire, KNX compounds them into one group address. Therefore, when it writes to the group address the KNXnet/IP driver must know the current value of every field before overwriting the DPT. To do this, the driver reads the whole group address before individually overwriting the data. This behaviour can be seen in the following ETS Group Monitor diagnostic of Group Address 7/1/0, which is a 222.100 DPT:

Figure 11 Datapoint Type compound structures



Adresse de Groupe: 7/1/0 Data point type: 222.100 room te Valeur: 0 0 0 0 0 0 ☐ Envoyer cycliquement Délai (sec): 0 Valeur reçue: 0C 1A 06 A4 05 DC

#	Heure	Service	Indicateurs	Prio	Source.adr	Source	Dest.adr	Dest	Rout	Type	DPT	Info
1	2016-12-08 13:41:00.757	du bus		Normal	15.15.255	-	7/1/0	Consigne Chaud Bât 4	6	Lecture		
2	2016-12-08 13:41:00.859	du bus		Normal	1.2.19	1.2.19 HALL V3000	7/1/0	Consigne Chaud Bât 4	6	Réponse	222.100 room temperature setpoint 00 00 00 00 00 00	
3	2016-12-08 13:41:00.905	du bus		Normal	15.15.255	-	7/1/0	Consigne Chaud Bât 4	6	Ecrire	222.100 room temperature setpoint 0C 1A 00 00 00 00	
4	2016-12-08 13:41:09.362	du bus		Normal	15.15.255	-	7/1/0	Consigne Chaud Bât 4	6	Lecture		
5	2016-12-08 13:41:09.464	du bus		Normal	1.2.19	1.2.19 HALL V3000	7/1/0	Consigne Chaud Bât 4	6	Réponse	222.100 room temperature setpoint 0C 1A 00 00 00 00	
6	2016-12-08 13:41:09.494	du bus		Normal	15.15.255	-	7/1/0	Consigne Chaud Bât 4	6	Ecrire	222.100 room temperature setpoint 0C 1A 06 A4 00 00	
7	2016-12-08 13:41:14.309	du bus		Normal	15.15.255	-	7/1/0	Consigne Chaud Bât 4	6	Lecture		
8	2016-12-08 13:41:14.467	du bus		Normal	1.2.19	1.2.19 HALL V3000	7/1/0	Consigne Chaud Bât 4	6	Réponse	222.100 room temperature setpoint 0C 1A 06 A4 00 00	
9	2016-12-08 13:41:14.495	du bus		Normal	15.15.255	-	7/1/0	Consigne Chaud Bât 4	6	Ecrire	222.100 room temperature setpoint 0C 1A 06 A4 05 DC	
10	2016-12-08 13:41:47.483	du bus		Normal	15.15.255	-	7/1/0	Consigne Chaud Bât 4	6	Lecture		
11	2016-12-08 13:41:47.581	du bus		Normal	1.2.19	1.2.19 HALL V3000	7/1/0	Consigne Chaud Bât 4	6	Réponse	222.100 room temperature setpoint 0C 1A 06 A4 05 DC	

- Line 1: Read group address

- Line 2: Response data `00 00 00 00 00 00`
- Line 3: Write TempSetpComf data `0C 1A 00 00 00 00` (Decimal 3098)
- Line 4: Read group address
- Line 5: Response data `0C 1A 00 00 00 00`
- Line 6: Write TempSetpStdby Data `0C 1A 06 A4 00 00` (Decimal 1700)
- Line 7: Read group address
- Line 8: Response data `0C 1A 06 A4 00 00`
- Line 9: Write TempSetpEco Data `0C 1A 06 A4 05 DC` (Decimal 1500)
- Line 10: Read group address
- Line 11: Response data `0C 1A 06 A4 05 DC`

Quick start

The basic steps to configure a station to communicate with a KNX system involve setting up Workbench, commissioning the Supervisor or JACE platform and configuring input proxy points.

Step 1 Plan and then configure the KNX system using the ETS (Engineering Tool Software).

This software is separate from the Niagara Workbench

Step 2 Using ETS, export the ETS Project. This provides the source of KNX system data for the KNXnet/IP driver.

Step 3 Copy the KNXnet/IP driver modules to the `C:\Niagara\Framework-n.n.nn\Modules` folder, where `Framework-n.n.nn` is the version of the

Step 4 Commission the remote controller platform.

Commissioning is documented in the controller guide.

Step 5 Set up a KnxNetwork in the station.

The detailed procedures in this guide begin with this step.

Step 6 Update the KNX Data Defs (data definitions).

Step 7 Configure the connection.

Step 8 Set up one or more KNX devices in the station.

Step 9 Set up one or more KNX points in the station.

Step 10 Set up alarms and other components.

For information on how to set up alarms, refer to *Niagara Alarms Guide*.

Chapter 2 Station setup

Topics covered in this chapter

- ♦ Adding a Knx Network
- ♦ About KNX Data Defs
- ♦ Configuring the connection to the KNX Network
- ♦ Device setup
- ♦ Adding points from an ETS project file
- ♦ Troubleshooting

The KNXnet/IP driver integrates a KNX system into a station.

The KNXnet/IP driver uses a familiar framework driver hierarchy of an upper-tier parent network component and one or more child device components, each with device extension (ext) child components.

Adding a Knx Network

Adding the network is the first step to configuring the station.

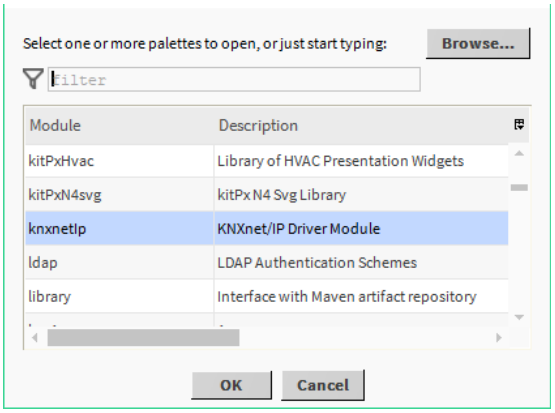
Prerequisites:

You configured and exported the ETS Project, copied the Knxnet/IP driver modules to the `Niagara` folder on the PC, commissioned the remote controller, and are working in Workbench on a PC.

Step 1 In the Nav tree, expand the station and double-click on **Drivers** node.

The **Driver Manager** view opens.

Step 2 In the **Driver Manager** view, open the KNXnet/IP driver palette in the side bar.

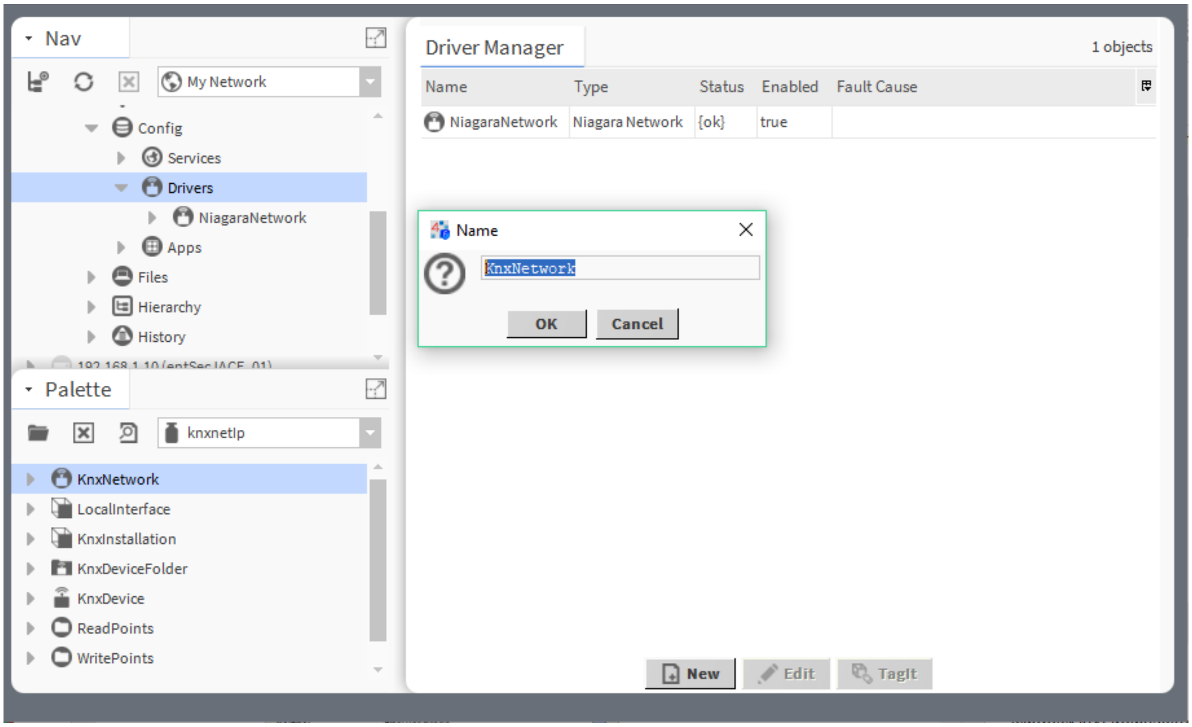


Select one or more palettes to open, or just start typing:

Module	Description
kitPxHvac	Library of HVAC Presentation Widgets
kitPxN4svg	kitPx N4 Svg Library
knxnetIp	KNXnet/IP Driver Module
ldap	LDAP Authentication Schemes
library	Interface with Maven artifact repository

Step 3 Select the **knxnetIp** module and click **OK**.

Step 4 Drag or copy the KnxNetwork component from the palette to the **Driver Manager** view.



Name	Type	Status	Enabled	Fault Cause
NiagaraNetwork	Niagara Network	{ok}	true	

Name: KnxNetwork

Step 5 Name the network and click **OK**.

The added KnxNetwork appears with a Status of `{down,alarm,unackedAlarm}`, because its **KNX Data Defs** have not yet been loaded.

You are now ready to load the data definitions.

About KNX Data Defs

For the KNXnet/IP driver to translate the encoded native KNX Group Address data from the KNX devices to the human friendly format displayed in the properties of control points, and vice versa, the driver needs information about the standard KNX Datapoint Types as published in the KNX specification document: `03_07_02 Datapoint Types v01.08.03 AS.pdf` (or later version). This information is collectively referred to as the KNX Data Definitions or KNX Data Defs.

The KNX Data Defs are stored in a file called knx_extra.xml. In addition, there is information about the units, minimum and maximum values, and compatible Control Point types for each of the KNX Datapoint Types Workbench contains a copy of the KNX Data Defs. Each station also has its own copy of KNX Data Defs.

The `knx_extra.xml` file works along similar lines to KNX's `knx_master.xml` file, which is part of the ETS installation and is updated from time to time with new KNX Datapoint Types by the KNX organization. The `knx_extra.xml` file is fundamental to the correct operation of the KNXnet/IP driver and it is internally signed to avoid the possibility of introducing erroneous operation of the KNXnet/IP driver.

Each release of the KNXnet/IP driver contains a version of `knx_extra.xml` that is current at the time of release, however newer versions of the `knx_extra.xml` file will be published to keep pace with changes to the KNX `knx_master.xml`.

NOTE: There is no correlation between the version number of the KNX Data Defs (for example `027`) and the version number of the KNXnet/IP driver module (for example `3.8.41.1.0.29`).

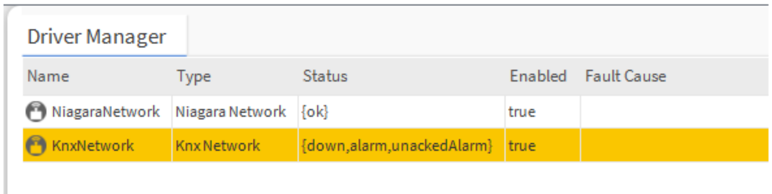
Loading the Data Defs

Loading the Knx Data Defs is a prerequisite to setting up the network. Workbench and the station maintain separate copies of data definitions. Each should contain the same definitions. This procedure assumes you are loading definitions for the first time into both Workbench and a remote controller station.

Prerequisites: You are working in Workbench on a PC. The KNX network driver has been added to the **Drivers** node in the appropriate station.

Step 1 In the station, expand **Station→Config** and double-click the **Drivers** node in the Nav tree.

The **Driver Manager** view opens.

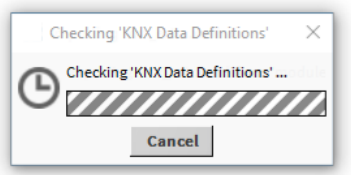


Name	Type	Status	Enabled	Fault Cause
NiagaraNetwork	Niagara Network	{ok}	true	
KnxNetwork	Knx Network	{down,alarm,unackedAlarm}	true	

Notice that the KnxNetwork is `{down}`.

Step 2 Double-click on the KNXNetwork driver row in the table.

If this is the first time you are using Workbench with a KnxNetwork, the driver checks for the KNX data definitions. If it detects new definitions, it updates the station database automatically.


Checking 'KNX Data Definitions'

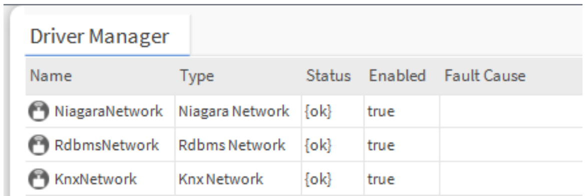
Checking 'KNX Data Definitions' ...

The driver makes only a single attempt to synchronize data definitions per session, unless the previous attempt failed.

After updating data definitions, it opens the **KNX Device Manager** view.

Step 3 Double-click the **Drivers** node in the Nav tree again.

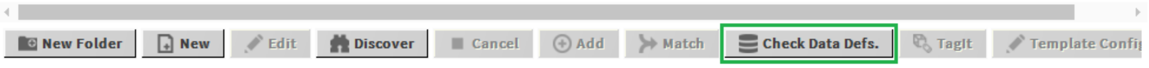
Once the data definitions are loaded, unless an error occurred, KNX Data Definition setup is complete, and the **Driver Manager** view clears the `{down}` status of the **KNX Network**.



Name	Type	Status	Enabled	Fault Cause
NiagaraNetwork	Niagara Network	{ok}	true	
RdbmsNetwork	Rdbms Network	{ok}	true	
KnxNetwork	Knx Network	{ok}	true	

Step 4 If the load failed, try one of the following:

- Open the **Knx Device Manager** (by double-clicking the **KnxNetwork** node in the Nav tree) or open the **Point Manager** (by expanding a **KnxNetwork→KnxDevice** and double-click the **Points** folder), then force a load attempt by clicking **Check Data Defs**.



- Click **Tools→KNX Import Service**, right-click **Knx Data Defs** and click the **Actions→Load Data Defs**.

The data definitions should load.

You are ready to update data definitions in the remote station and configure the KNX network driver's connection to the physical KNXnet/IP network.

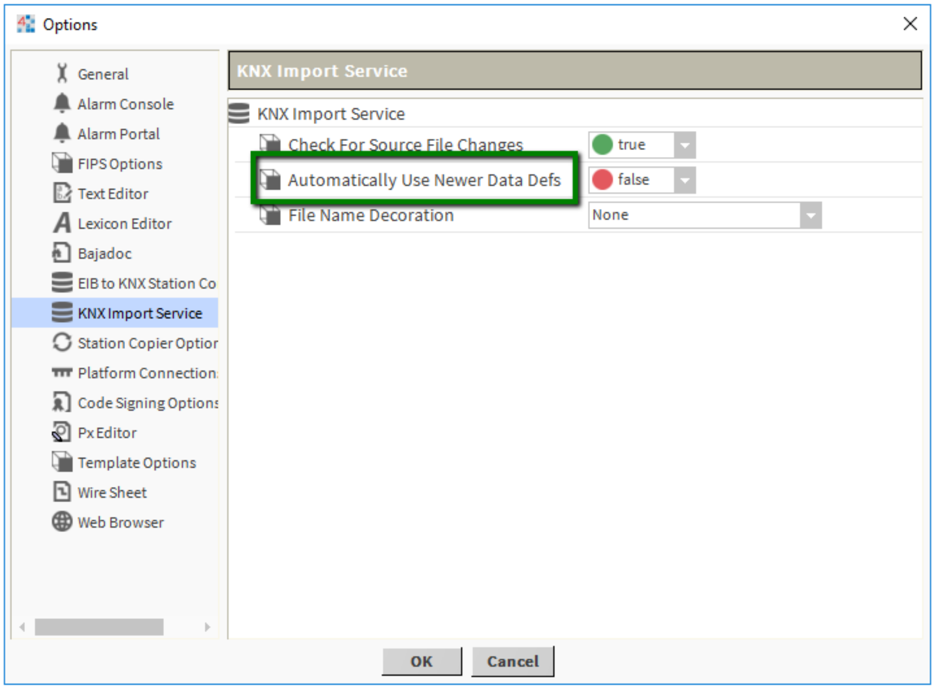
Loading the Data Defs manually

If the automatic load of the KNX Data Defs failed, you can load them manually.

Prerequisites: You are working in Workbench with the **Driver Manager** view open.

Step 1 Expand **Tools→Options→KNX Import Service**.

Step 2 Set **`Automatically Use Newer Data Defs`** to `false`.

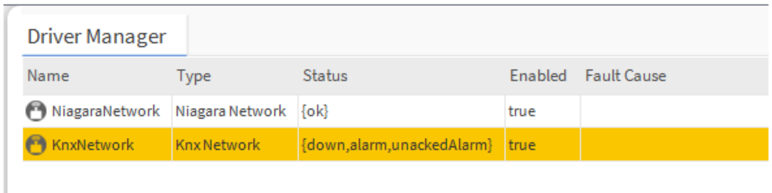


KNX Import Service

KNX Import Service

Check For Source File Changes	true
Automatically Use Newer Data Defs	false
File Name Decoration	None

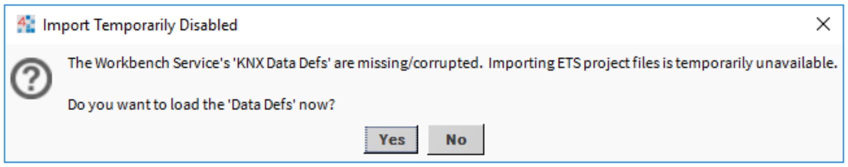
Step 3 In the **Driver Manager** view, double-click on the **KNX Network driver** and open its **Device Manager** view.



Driver Manager

Name	Type	Status	Enabled	Fault Cause
NiagaraNetwork	Niagara Network	{ok}	true	
KnxNetwork	Knx Network	{down,alarm,unackedAlarm}	true	

Step 4 If this is the first time your Workbench has been used with a **KNX Network**, the **Import Temporarily Disabled** window opens.

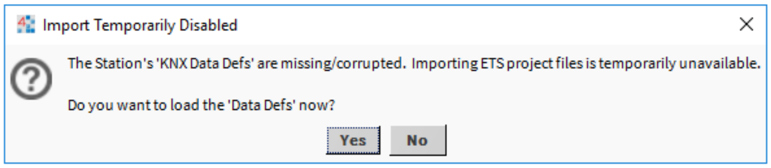


The Workbench Service's 'KNX Data Defs' are missing/corrupted. Importing ETS project files is temporarily unavailable.

Do you want to load the 'Data Defs' now?

Step 5 To load the Workbench Service's **KNX Data Defs**, click **Yes**.

The **Import Temporarily Disabled** window, with reference to the station's **KNX Data**, opens.

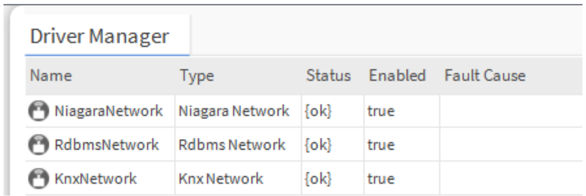


The Station's 'KNX Data Defs' are missing/corrupted. Importing ETS project files is temporarily unavailable.

Do you want to load the 'Data Defs' now?

Step 6 To load the station’s **KNX Data Defs**, click **Yes**.

The **Driver Manager** view clears the `{down}` status of the **KNX Network**.

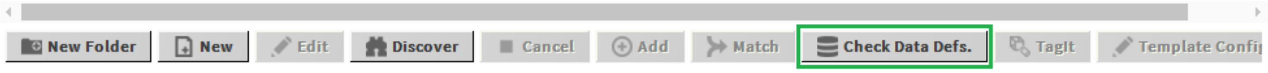


Name	Type	Status	Enabled	Fault Cause
NiagaraNetwork	Niagara Network	{ok}	true	
RdbmsNetwork	Rdbms Network	{ok}	true	
KnxNetwork	Knx Network	{ok}	true	

You are now ready to configure the KNX network driver’s connection to the physical KNXnet/IP network.

NOTE:

If loading failed for some reason, there is a button available in the **Device** and **Point Manager** views to force a loading attempt.



Examining the Data Defs

You can examine the current status and version of the KNX Data Defs in use in both the Supervisor and remote stations.

Prerequisites: You are connected to the station and working in Workbench

Step 1 To view the Data Defs in the Supervisor station, expand **Tools** and double-click the **KNX Import Service**.

The **KNX Import Service** view opens.

Step 2 Expand Knx Data Defs and review the definitions that are currently loaded in the Supervisor station.

Step 3 To view the Data Defs in a remote controller station, expand **Config→Drivers** in the Nav tree, right-click the **Knx Network** and click **Views→AX Property Sheet**.

The network **Property Sheet** opens.

Step 4 Expand Knx Data Defs and review the definitions that are currently loaded in the remote station.

Updating the Data Defs

From time to time the KNX Data Defs in the Workbench and station may need to be updated. Workbench and the station maintain separate copies of data definitions. This procedure updates the Workbench definitions and uses them to update the remote station’s data definitions.

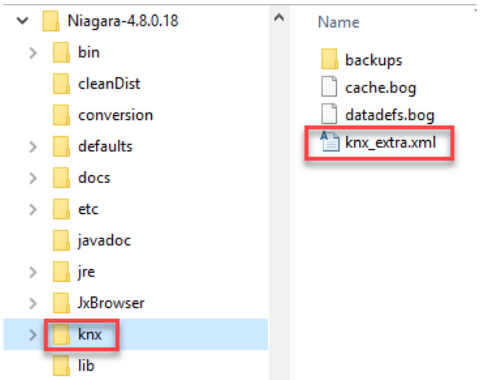
Prerequisites: The PC running Workbench and the remote station already contain KNX Data Defs.

New versions of the `knx_extra.xml` file are published to a `DropBox` folder available here: `https://www.dropbox.com/sh/uwkdggd10fi1955/AAAhMkD6CoQtLx-A_DPEoguga?dl=0`

The latest version is always `knx_extra.xml` in the root of this shared `Dropbox` folder.

NOTE: Older versions are available in the `Previous Versions of knx_extra.xml` sub-folder, however it should not be necessary to use these and if one were required you must rename it to remove the `_vXXX` suffix.

Step 1 Obtain the latest `knx_extra.xml` from `Dropbox` and put it in the `knx` sub-folder of your Niagara installation on your PC (for example:. `C:\Niagara\Niagara-n.n.nn\knx\knx_extra.xml`).

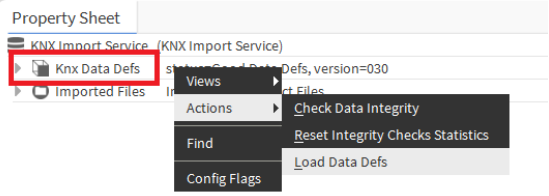


Step 2 In Workbench, click **Tools→KNX Import Service**.

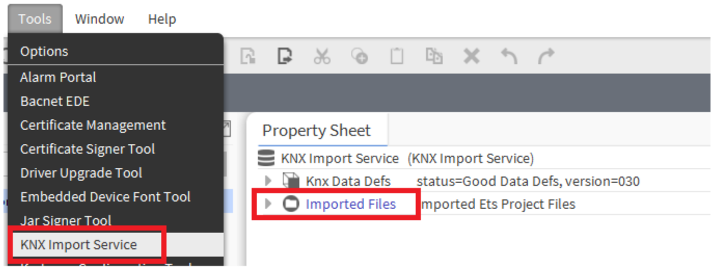
The AX Property Sheet opens.

Step 3 Do one of the following: To automatically update the Workbench data definitions,

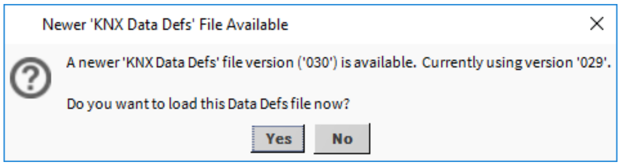
- Right-click **`Knx Data Defs`** followed by clicking the **Actions→Load Data Defs**.



- Click **`Imported Files`**.



The driver examines the new `knx_extra.xml` file you copied into the `knx` sub-folder of your installation and displays:


Newer 'KNX Data Defs' File Available

A newer 'KNX Data Defs' file version ('030') is available. Currently using version '029'.

Do you want to load this Data Defs file now?

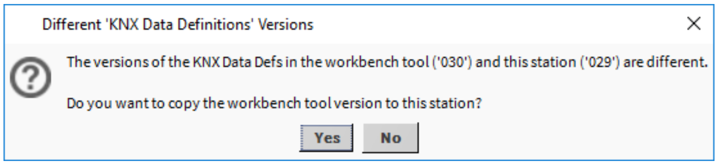
Step 4 To continue with the automatic update, click **Yes**.

The driver automatically updates the data definitions.

If you clicked `Imported Files`, the **Ets Project File Import Manager** view opens. Otherwise the focus remains on the **AX Property Sheet**.

Step 5 To update the remote station's KNX Data Defs, navigate to the Supervisor's **Knx Device Manager** or **Knx Point Manager** view.

As it loads the view, the driver checks the station's version of the data definitions against Workbench's version. Since the versions differ, the driver prompts you:



Different 'KNX Data Definitions' Versions

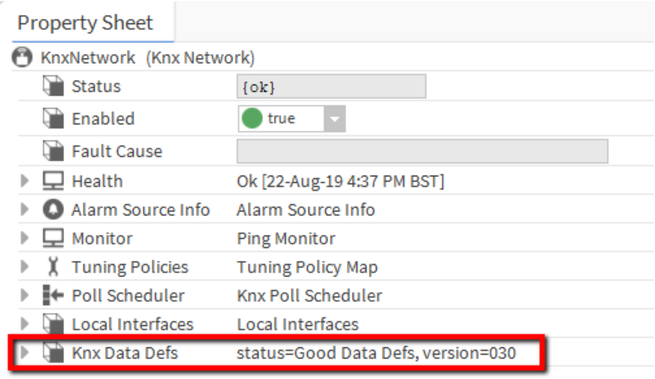
The versions of the KNX Data Defs in the workbench tool ('030') and this station ('029') are different.

Do you want to copy the workbench tool version to this station?

Step 6 To synchronize data definitions between Workbench and the station, click **Yes**.

The driver updates the data definitions.

Step 7 To view the current status of the data definitions in the remote station, right-click the **KnxNetwork** node in the Nav tree, click **Views→AX Property Sheet**,and confirm the status of the `Knx Data Defs`.



Property Sheet

KnxNetwork (Knx Network)

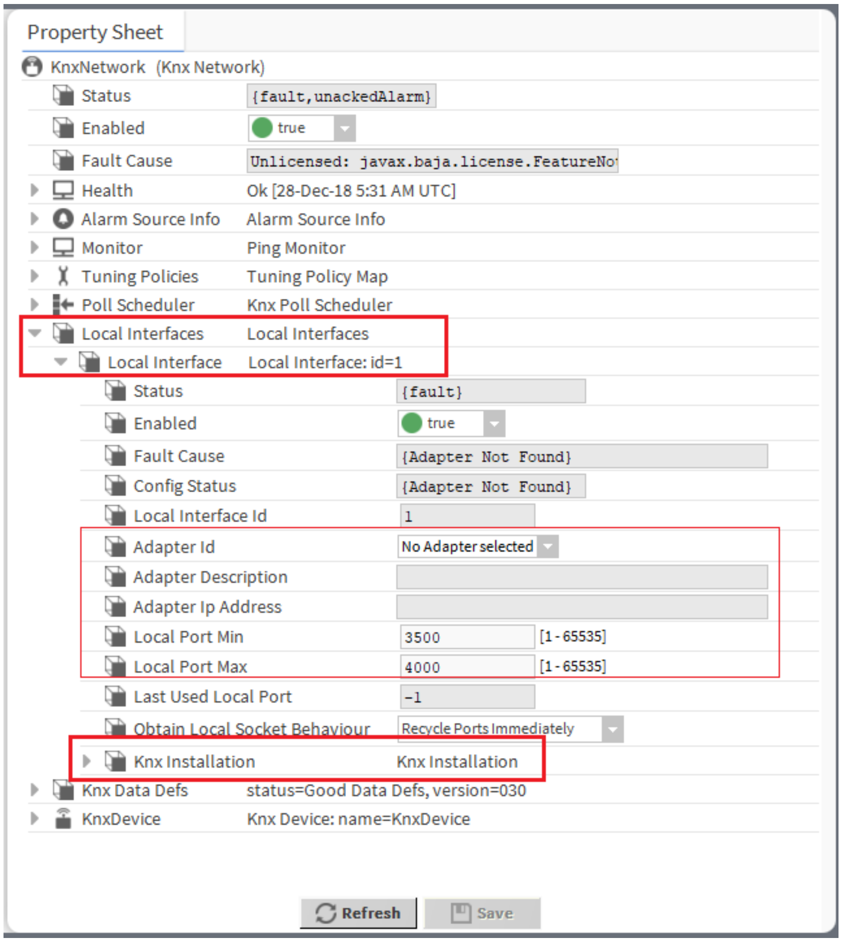
Status	{ok}
Enabled	true
Fault Cause	
Health	Ok [22-Aug-19 4:37 PM BST]
Alarm Source Info	Alarm Source Info
Monitor	Ping Monitor
Tuning Policies	Tuning Policy Map
Poll Scheduler	Knx Poll Scheduler
Local Interfaces	Local Interfaces
Knx Data Defs	status=Good Data Defs, version=030

Configuring the connection to the KNX Network

The KNXnet/IP driver's connection to the physical network can now be configured.

Prerequisites: The KNX network driver has been added to the **Drivers** node and the Data Defs loaded in Workbench and the station.

Step 1 To open the KnxNetwork **Property Sheet**, right-click the **KnxNetwork** node in the Nav tree, click **Views→AX Property Sheet**, and expand the **Local Interfaces→Local Interface** properties.



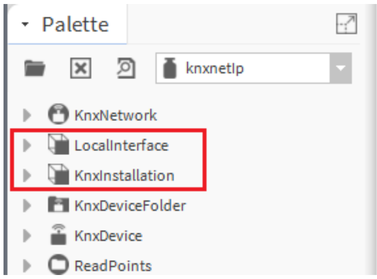
KnxNetwork (Knx Network)

Status	{fault,unackedAlarm}
Enabled	true
Fault Cause	Unlicensed: javax.baja.license.FeatureNo
Health	Ok [28-Dec-18 5:31 AM UTC]
Alarm Source Info	Alarm Source Info
Monitor	Ping Monitor
Tuning Policies	Tuning Policy Map
Poll Scheduler	Knx Poll Scheduler
Local Interfaces	Local Interfaces
Local Interface	Local Interface: id=1
Status	{fault}
Enabled	true
Fault Cause	{Adapter Not Found}
Config Status	{Adapter Not Found}
Local Interface Id	1
Adapter Id	No Adapter selected
Adapter Description	
Adapter Ip Address	
Local Port Min	3500 [1 - 65535]
Local Port Max	4000 [1 - 65535]
Last Used Local Port	-1
Obtain Local Socket Behaviour	Recycle Ports Immediately
Knx Installation	Knx Installation
Knx Data Defs	status=Good Data Defs, version=030
KnxDevice	Knx Device: name=KnxDevice

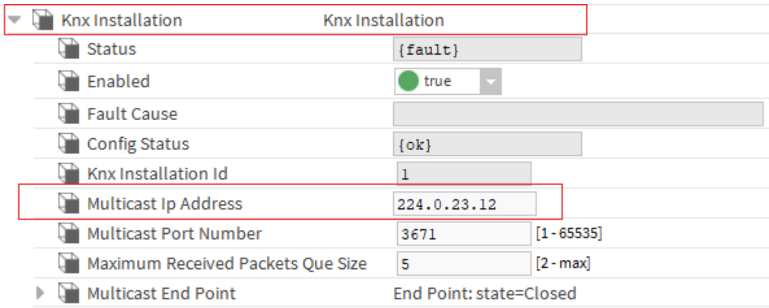
A default Local Interface has been added to the KNXnet/IP driver and a default Knx Installation has been added to the Local Interface.

Step 2 Select the **`Adapter Id`** to use for this connection to the physical KNX Network.

NOTE: You can add additional Local Interfaces from the palette for each of the platform's TCP/IP Interfaces to use for connecting to a physical KNX Network. All the Local Interfaces in the station must have different Adapter Ids.



Step 3 Expand the Knx Installation section of the **AX Property Sheet** and enter the **`Multicast Ip Address`**.



Knx Installation	Knx Installation	
Status	{fault}	
Enabled	true	
Fault Cause		
Config Status	{ok}	
Knx Installation Id	1	
Multicast Ip Address	224.0.23.12	
Multicast Port Number	3671	[1 - 65535]
Maximum Received Packets Que Size	5	[2 - max]
Multicast End Point	End Point: state=Closed	

NOTE: If you are using more than one Knx IP Multicast Address, you can add additional Knx Installation instances from the palette to any Local Interface. All the Knx Installation instances under a Local Interface must have different Knx IP Multicast Addresses.

Step 4 To update the database, click **Save**.

You are now ready to add KNXnet/IP interface devices to the KNX network driver.

Device setup

Setting up the network involves adding and configuring devices and points.

You add KNX device instances to the station in three ways:

- By using the framework to discover the devices on a connected, physical KNX network
- By importing device data from an ETS project file (`*.knxproj`)
- By manually inputting the device data

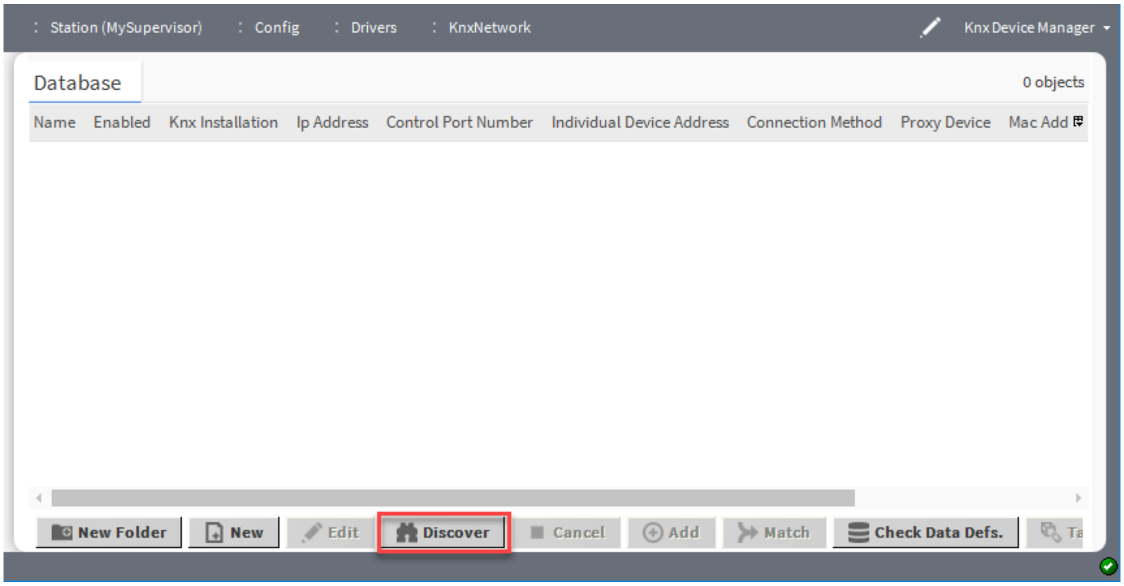
Discovering devices

Discovery adds KNX device instances to the station database.

Prerequisites: The KNX network driver has been be added to the **Drivers** node. The KNX Data Defs are loaded. The connection to the physical KNXnet/IP network has been configured.

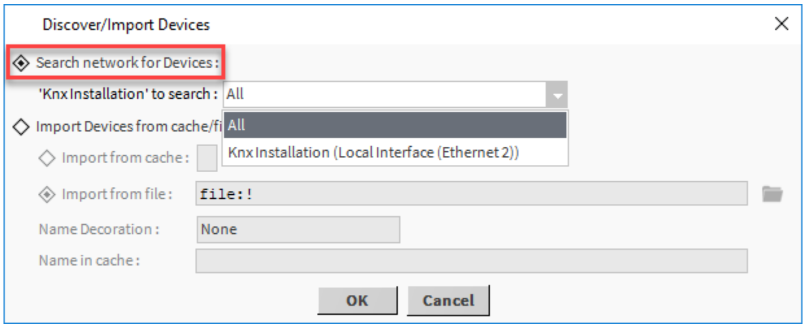
Step 1 Expand the station **Config** node, double-click the **Driver Manager** node, and double-click the KNXNetwork driver row in the table.

The **Knx Device Manager** view opens.



Step 2 Click **Discover**.

The **Discover/ImportDevices** window opens.


Discover/Import Devices
Search network for Devices :
Import Devices from cache/fi
Import from cache :
Import from file :
Name Decoration :
Name in cache :
All
Knx Installation (Local Interface (Ethernet 2))
OK
Cancel

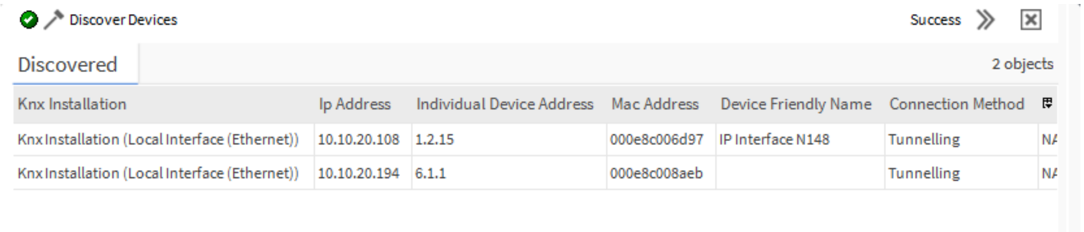
'Knx Installation' to search : All

Import from file : file:!

Name Decoration : None

Step 3 Select the `Search network for Devices` option, choose which Knx Installation to search from the drop-down list and click **OK**.

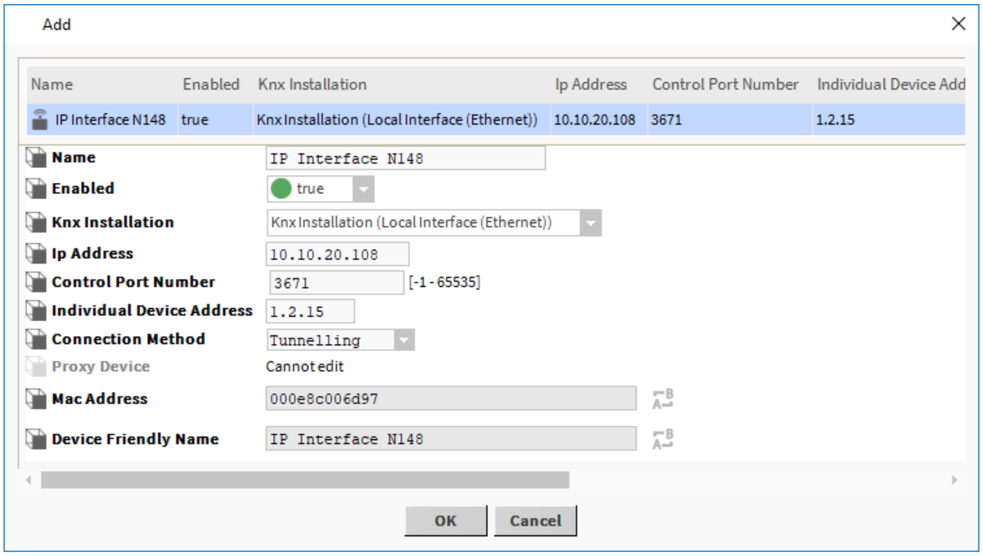
The discovery process takes about 10 seconds for each Knx Installation search.



Knx Installation	Ip Address	Individual Device Address	Mac Address	Device Friendly Name	Connection Method	
Knx Installation (Local Interface (Ethernet))	10.10.20.108	1.2.15	000e8c006d97	IP Interface N148	Tunnelling	N/
Knx Installation (Local Interface (Ethernet))	10.10.20.194	6.1.1	000e8c008aeb		Tunnelling	N/

Step 4 Drag Device(s) from the **Discovered** pane to the **Database** pane or click **Add**.

The **Add** window opens.



Name	Enabled	Knx Installation	Ip Address	Control Port Number	Individual Device Add
IP Interface N148	true	Knx Installation (Local Interface (Ethernet))	10.10.20.108	3671	1.2.15

Name	IP Interface N148
Enabled	true
Knx Installation	Knx Installation (Local Interface (Ethernet))
Ip Address	10.10.20.108
Control Port Number	3671 [-1 - 65535]
Individual Device Address	1.2.15
Connection Method	Tunnelling
Proxy Device	Cannot edit
Mac Address	000e8c006d97
Device Friendly Name	IP Interface N148

NOTE: As discovery adds the devices to the database it updates the **`Knx Installation`** and **`Ip Address`** properties. You do not need to configure them here.

Step 5 Make any other configuration changes and click **OK**.

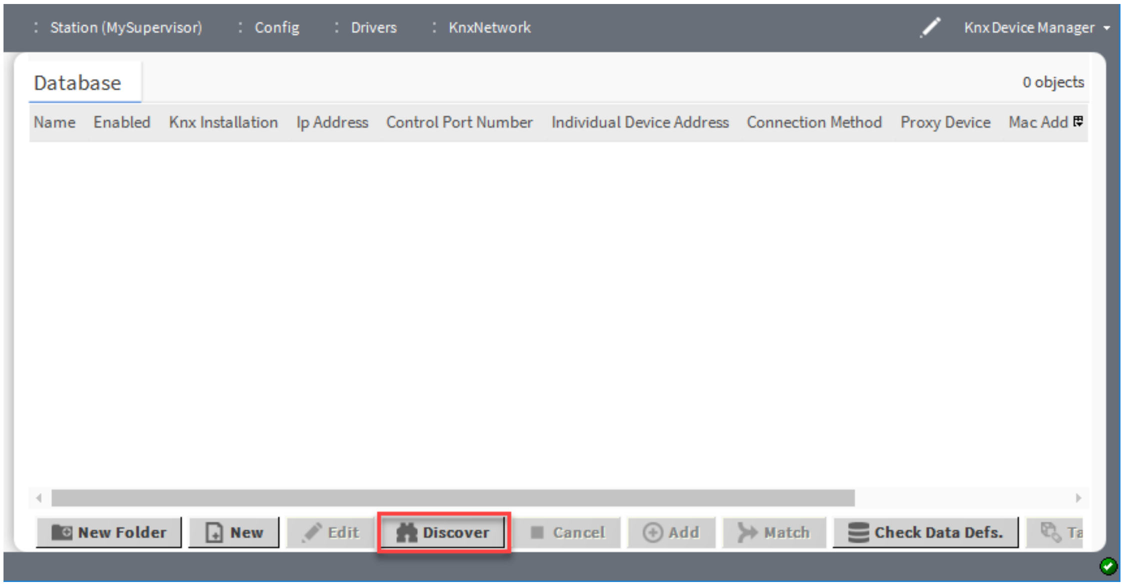
Importing devices from an ETS project file

Instead of discovering devices, you may import them from an ETS project file.

Prerequisites: The KNX network driver has been be added to the **Drivers** node. The KNX Data Defs are loaded. The connection to the physical KNXnet/IP network has been configured.

Step 1 Expand the station **Config** node, double-click the **Driver Manager** node, and double-click the KNXNetwork driver row in the table.

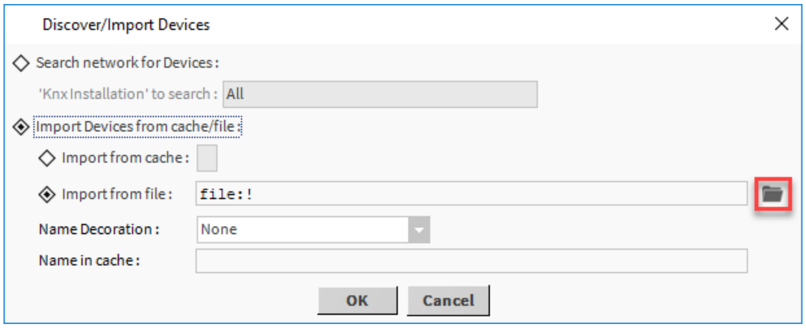
The **Knx Device Manager** view opens.



Name	Enabled	Knx Installation	Ip Address	Control Port Number	Individual Device Address	Connection Method	Proxy Device	Mac Add

Step 2 Click **Discover**.

The **Discover/Import Devices** window opens.



Discover/Import Devices

◇ Search network for Devices:

'Knx Installation' to search: All

◈ Import Devices from cache/file:

◇ Import from cache:

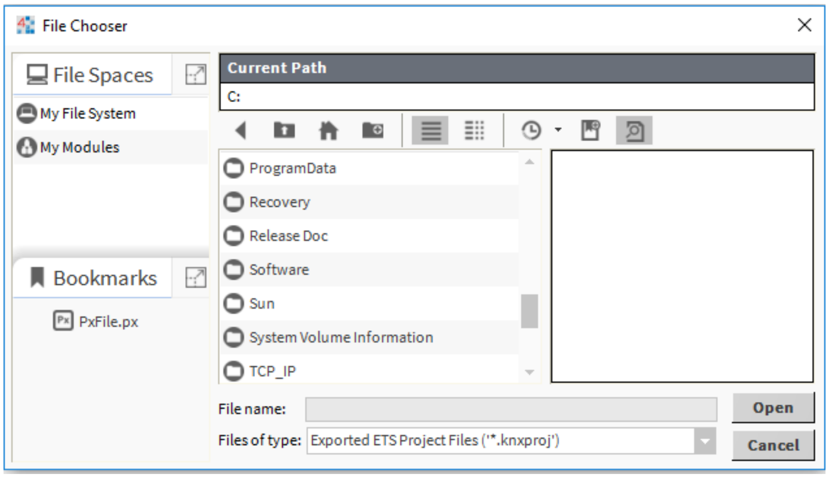
◈ Import from file: file:!

Name Decoration: None

Name in cache:

Step 3 Select the `Import Devices from cache/file` and the `Import from file` options and click on the File Chooser.

The **File Chooser** window opens.



File Chooser

Current Path: C:

- ProgramData
- Recovery
- Release Doc
- Software
- Sun
- System Volume Information
- TCP_IP

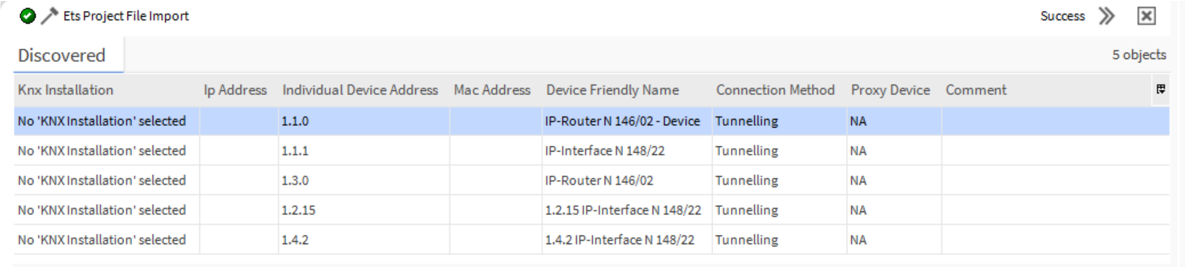
File name:

Files of type: Exported ETS Project Files ('*.knxproj')

Step 4 Navigate to and select your ETS Project file, click **Open**, then click **OK** in the **Discover/Import Devices** window.

The time it takes to import an **ETS Project file** is proportional to its size.

The **Ets Project File Import** view opens.

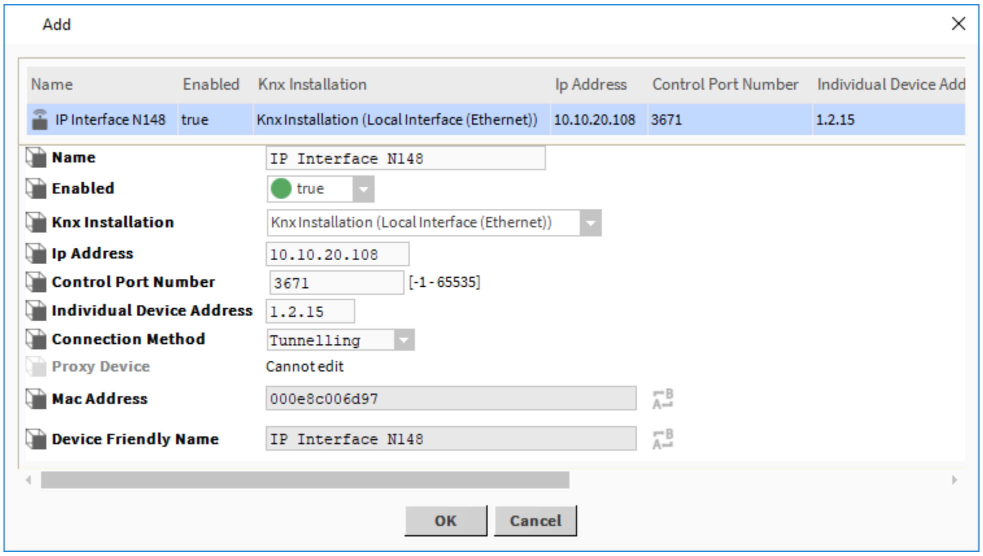


Ets Project File Import — Success

Discovered — 5 objects

Knx Installation	Ip Address	Individual Device Address	Mac Address	Device Friendly Name	Connection Method	Proxy Device	Comment
No 'KNX Installation' selected		1.1.0		IP-Router N 146/02 - Device	Tunnelling	NA	
No 'KNX Installation' selected		1.1.1		IP-Interface N 148/22	Tunnelling	NA	
No 'KNX Installation' selected		1.3.0		IP-Router N 146/02	Tunnelling	NA	
No 'KNX Installation' selected		1.2.15		1.2.15 IP-Interface N 148/22	Tunnelling	NA	
No 'KNX Installation' selected		1.4.2		1.4.2 IP-Interface N 148/22	Tunnelling	NA	

Step 5 Drag the device(s) from the **Discovered** pane to the **Database** pane or click **Add** to display the **Add** window.



Name	Enabled	Knx Installation	Ip Address	Control Port Number	Individual Device Add
IP Interface N148	true	Knx Installation (Local Interface (Ethernet))	10.10.20.108	3671	1.2.15

Name	IP Interface N148
Enabled	● true
Knx Installation	Knx Installation (Local Interface (Ethernet))
Ip Address	10.10.20.108
Control Port Number	3671 [-1 - 65535]
Individual Device Address	1.2.15
Connection Method	Tunnelling
Proxy Device	Cannot edit
Mac Address	000e8c006d97
Device Friendly Name	IP Interface N148

NOTE: This process imports only the **`Individual Device Address`** from the ETS Project file. You must configure the **`Knx Installation`** for each device. It is not necessary to configure the **`Ip Address`** now because the first time the framework connects to the device, using the device's **`Individual Device Address`**, it retrieves the **`Ip Address`** from the device itself.

Step 6 Configure the **`Knx Installation`** and click **OK**.

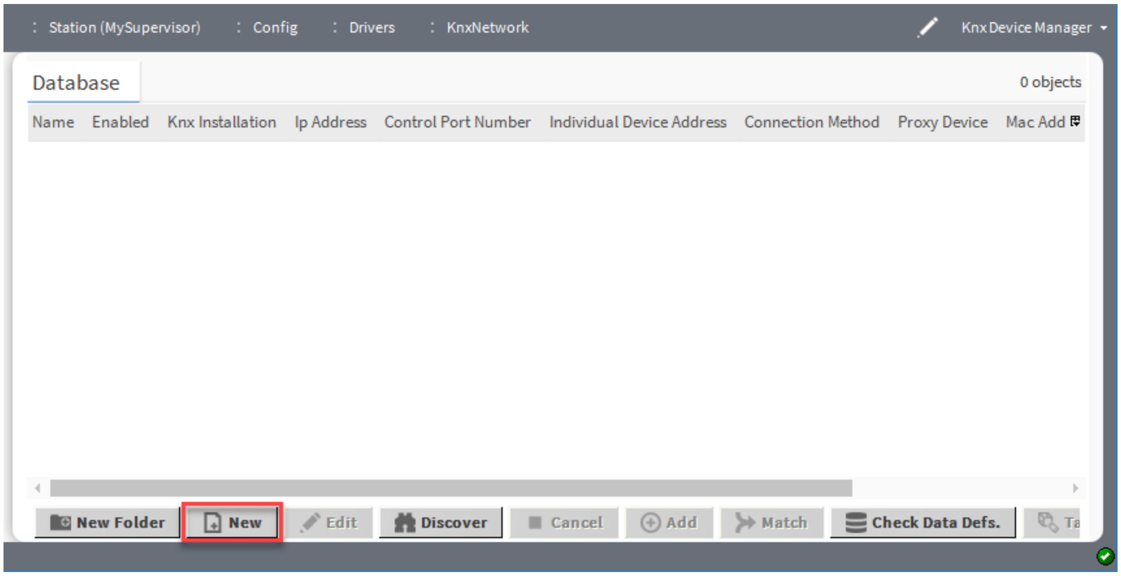
Adding devices manually

If neither discovery nor importing from the ETS project file works you can add devices manually.

Prerequisites: The KNX network driver has been be added to the **Drivers** node. The KNX Data Defs are loaded. The connection to the physical KNXnet/IP network has been configured.

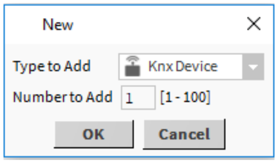
Step 1 Expand the station **Config** node, double-click the **Driver Manager** node, and double-click the KNXNetwork driver row in the table.

The **Device Manager** view opens.



Step 2 Click **New**.

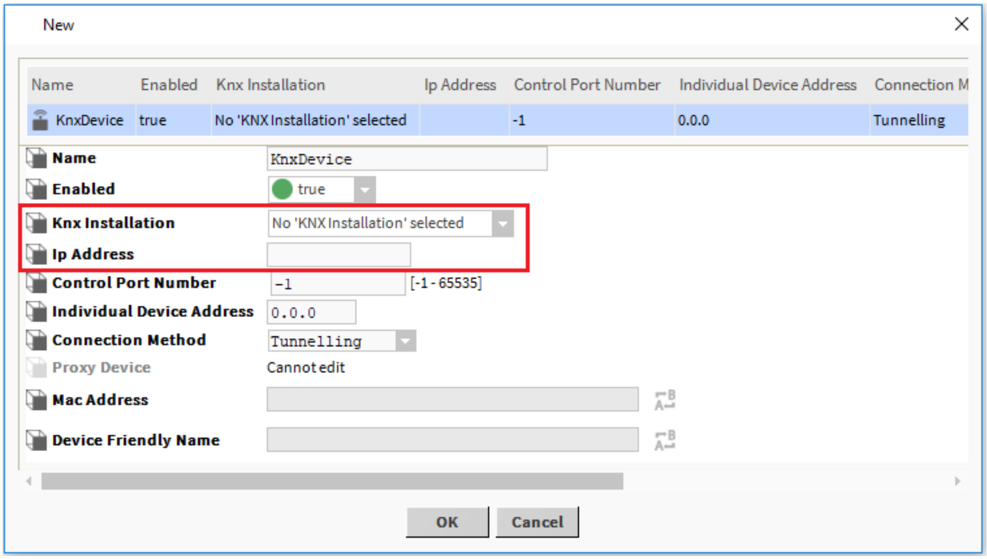
The **New** [devices] window opens.



Type to Add: Knx Device
Number to Add: 1 [1 - 100]

Step 3 Enter the **Number to Add** [of devices] and click **OK**.

A second **New** window opens.



Name	Enabled	Knx Installation	Ip Address	Control Port Number	Individual Device Address	Connection M
KnxDevice	true	No 'KNX Installation' selected		-1	0.0.0	Tunnelling

Name: KnxDevice
Enabled: true
Knx Installation: No 'KNX Installation' selected
Ip Address:
Control Port Number: -1 [-1 - 65535]
Individual Device Address: 0.0.0
Connection Method: Tunnelling
Proxy Device: Cannot edit
Mac Address:
Device Friendly Name:

NOTE: When you add devices manually, you must configure both the **`Knx Installation`** and **`Ip Address`** for each device. It is not necessary to configure the **`Individual Device Address`** because, when the KnxNetwork driver first tries to connect to the device, the driver uses the **`Ip Address`** to discover the **`Individual Device Address`** from the dervice itself.

Step 4 Configure any other properties and click **OK**.

Adding points from an ETS project file

The only method to add KNX points through discovery is by using the ETS project file.

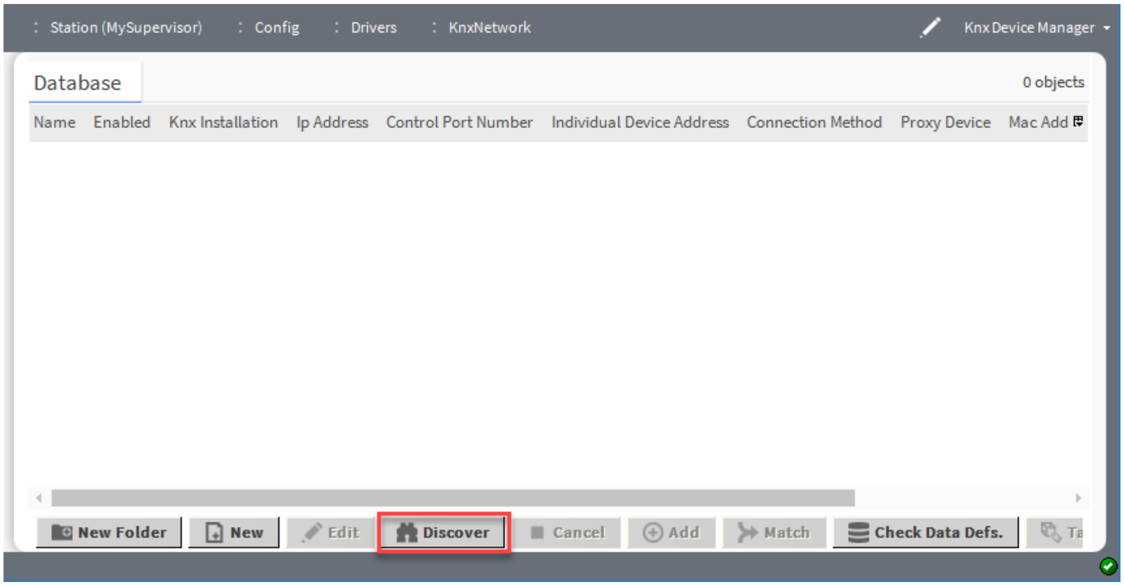
Prerequisites:

One or more Knx device instances has been be added and configured to the Knxnet/IP driver.

Step 1 Expand **Config→Drivers→KnxNetwork→KnxDevice**.

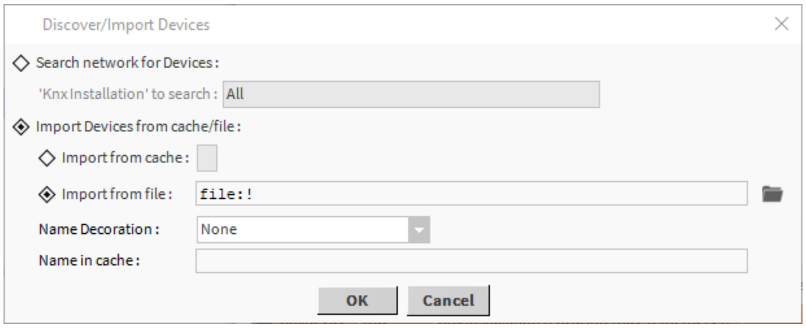
Step 2 Double-click on **Points** node.

The **Knx Point Manager** view opens.



Step 3 Click **Discover**.

The **Discover/Import Devices** window opens.


Discover/Import Devices

◇ Search network for Devices :

'Knx Installation' to search : All

◈ Import Devices from cache/file :

◇ Import from cache : ☐

◈ Import from file : file:!

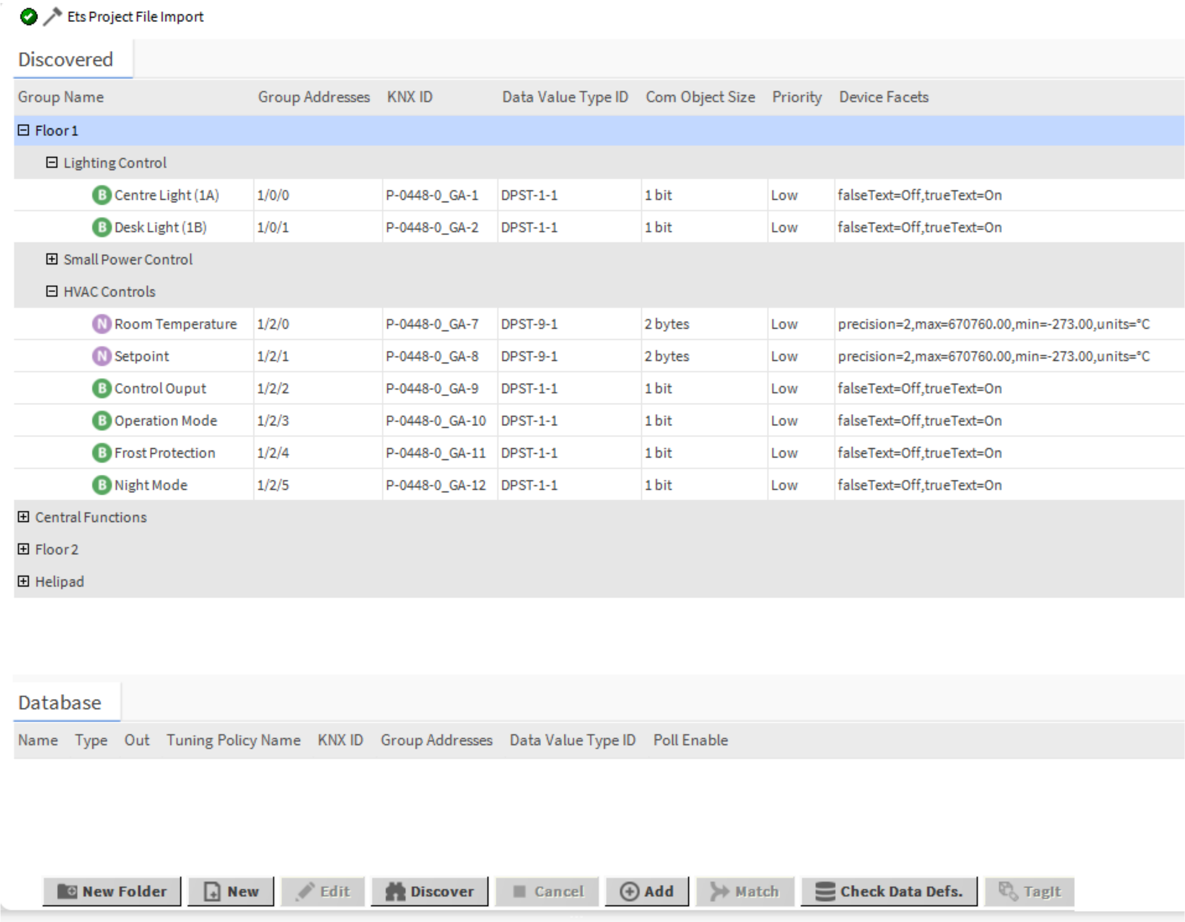
Name Decoration : None

Name in cache :

NOTE: The cache retains all the previously loaded ETS project files. To speed import, you can import from cache. When you do so, the driver checks to detect changes to the original file since it was last imported into the cache.

Step 4 Select `Import from cache` or `Import from file` option and click **OK**.

The **Ets Project File Import** view opens.



Discovered

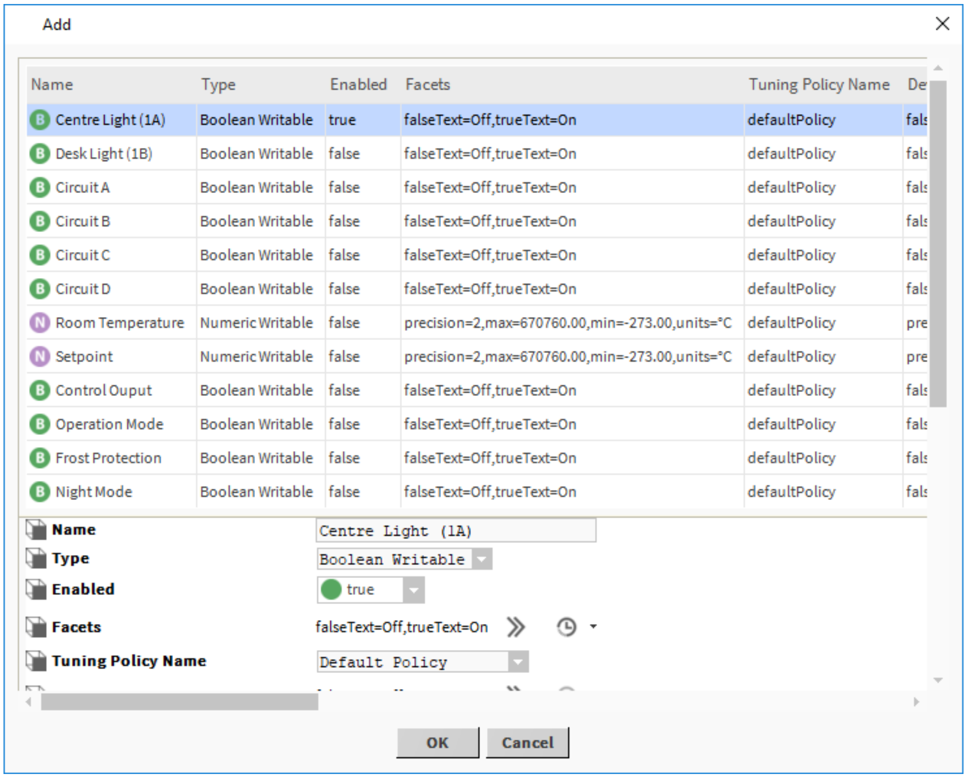
Group Name	Group Addresses	KNX ID	Data Value Type ID	Com Object Size	Priority	Device Facets
⊟ Floor 1						
⊟ Lighting Control						
B Centre Light (1A)	1/0/0	P-0448-0_GA-1	DPST-1-1	1 bit	Low	falseText=Off,trueText=On
B Desk Light (1B)	1/0/1	P-0448-0_GA-2	DPST-1-1	1 bit	Low	falseText=Off,trueText=On
⊞ Small Power Control						
⊟ HVAC Controls						
N Room Temperature	1/2/0	P-0448-0_GA-7	DPST-9-1	2 bytes	Low	precision=2,max=670760.00,min=-273.00,units=°C
N Setpoint	1/2/1	P-0448-0_GA-8	DPST-9-1	2 bytes	Low	precision=2,max=670760.00,min=-273.00,units=°C
B Control Ouput	1/2/2	P-0448-0_GA-9	DPST-1-1	1 bit	Low	falseText=Off,trueText=On
B Operation Mode	1/2/3	P-0448-0_GA-10	DPST-1-1	1 bit	Low	falseText=Off,trueText=On
B Frost Protection	1/2/4	P-0448-0_GA-11	DPST-1-1	1 bit	Low	falseText=Off,trueText=On
B Night Mode	1/2/5	P-0448-0_GA-12	DPST-1-1	1 bit	Low	falseText=Off,trueText=On
⊞ Central Functions						
⊞ Floor 2						
⊞ Helipad						

Database

Name	Type	Out	Tuning Policy Name	KNX ID	Group Addresses	Data Value Type ID	Poll Enable

Step 5 Drag point(s) from the **Discovered** pane into the **Database** pane or click **Add**.

The **Add** window opens.



Name	Type	Enabled	Facets	Tuning Policy Name	De
Centre Light (1A)	Boolean Writable	true	falseText=Off,trueText=On	defaultPolicy	fals
Desk Light (1B)	Boolean Writable	false	falseText=Off,trueText=On	defaultPolicy	fals
Circuit A	Boolean Writable	false	falseText=Off,trueText=On	defaultPolicy	fals
Circuit B	Boolean Writable	false	falseText=Off,trueText=On	defaultPolicy	fals
Circuit C	Boolean Writable	false	falseText=Off,trueText=On	defaultPolicy	fals
Circuit D	Boolean Writable	false	falseText=Off,trueText=On	defaultPolicy	fals
Room Temperature	Numeric Writable	false	precision=2,max=670760.00,min=-273.00,units=°C	defaultPolicy	pre
Setpoint	Numeric Writable	false	precision=2,max=670760.00,min=-273.00,units=°C	defaultPolicy	pre
Control Ouput	Boolean Writable	false	falseText=Off,trueText=On	defaultPolicy	fals
Operation Mode	Boolean Writable	false	falseText=Off,trueText=On	defaultPolicy	fals
Frost Protection	Boolean Writable	false	falseText=Off,trueText=On	defaultPolicy	fals
Night Mode	Boolean Writable	false	falseText=Off,trueText=On	defaultPolicy	fals

Name	Centre Light (1A)
Type	Boolean Writable
Enabled	true
Facets	falseText=Off,trueText=On
Tuning Policy Name	Default Policy

Step 6 To add points into the database, click **OK**.

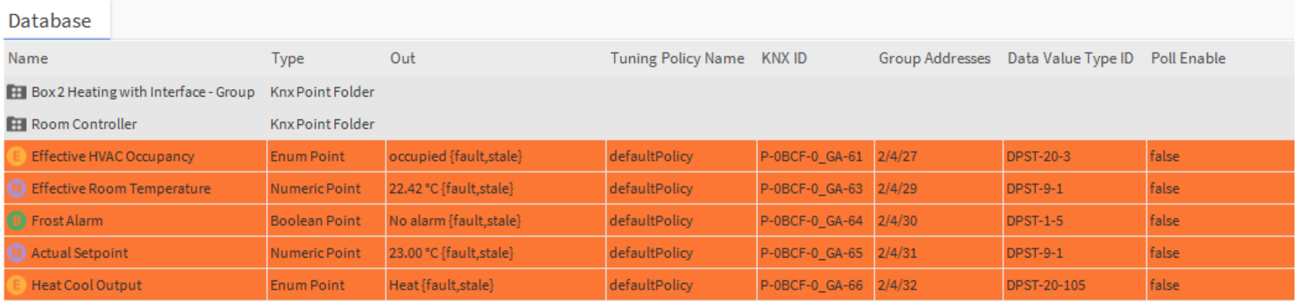
Troubleshooting

This topic covers some issues and how to resolve them.

Can I have both the KNXnet/IP driver and EIBnet/IP driver running in my station?

No. There is an absolute certainty of conflict between the KNXnet/IP driver and the older EIBnet/IP driver if they are running concurrently. If there is an EIBnet/IP driver in the station, delete it before you enable the KNXnet/IP driver.

Why, when discovered points are added to the Database, are they in {fault}?

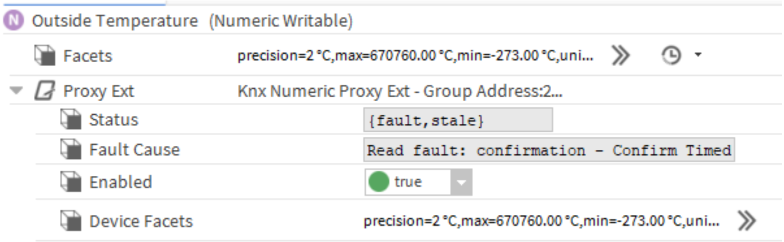


Name	Type	Out	Tuning Policy Name	KNX ID	Group Addresses	Data Value Type ID	Poll Enable
Box 2 Heating with Interface - Group	Knx Point Folder						
Room Controller	Knx Point Folder						
Effective HVAC Occupancy	Enum Point	occupied {fault,stale}	defaultPolicy	P-0BCF-0_GA-61	2/4/27	DPST-20-3	false
Effective Room Temperature	Numeric Point	22.42 °C {fault,stale}	defaultPolicy	P-0BCF-0_GA-63	2/4/29	DPST-9-1	false
Frost Alarm	Boolean Point	No alarm {fault,stale}	defaultPolicy	P-0BCF-0_GA-64	2/4/30	DPST-1-5	false
Actual Setpoint	Numeric Point	23.00 °C {fault,stale}	defaultPolicy	P-0BCF-0_GA-65	2/4/31	DPST-9-1	false
Heat Cool Output	Enum Point	Heat {fault,stale}	defaultPolicy	P-0BCF-0_GA-66	2/4/32	DPST-20-105	false

One of the many reasons why points can go to a `{fault}` condition, relates to the R1 firmware revision of the Siemens N 148/22 KNXnet/IP Interface device. It may also occur with other KNXnet/IP Interface devices.

The problem is caused by the KNXnet/IP Interface device failing to respond with an acknowledge message although message confirmation has been requested by the KNXnet/IP driver.

You may also observe that the **Fault Cause** property of the point proxy extension indicates `Read fault: confirmation – Confirm Timed Out`.

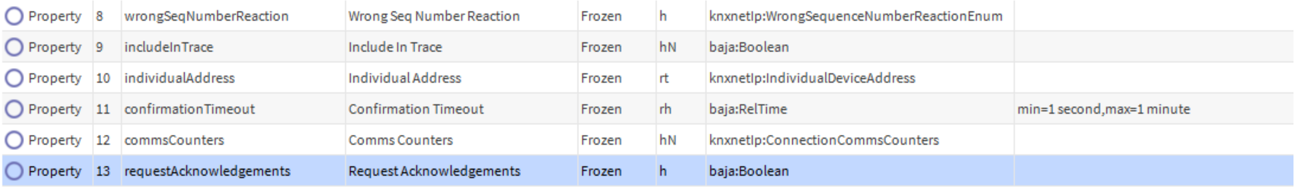


Outside Temperature (Numeric Writable)	
Facets	precision=2 °C,max=670760.00 °C,min=-273.00 °C,uni...
Proxy Ext	Knx Numeric Proxy Ext - Group Address:2...
Status	{fault,stale}
Fault Cause	Read fault: confirmation - Confirm Timed
Enabled	true
Device Facets	precision=2 °C,max=670760.00 °C,min=-273.00 °C,uni...

One way to overcome this problem is to upgrade the R1 firmware revision level of the Siemens N 148/22 KNXnet/IP Interface device (or any other KNXnet/IP interface device). It is beyond the scope of this document to detail the steps to accomplish this, but manufacturers of KNXnet/IP interface devices may provide tools and guidance to upgrade their firmware.

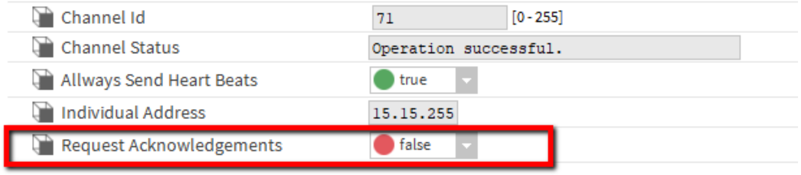
Another way to overcome the problem is available within the KNXnet/IP driver. To do this, configure the KNXnet/IP driver not to request an acknowledgement to its messages. To configure this use a hidden property in the tunnel connection of the KNXnet/IP driver.

Unhide the **requestAcknowledgements** slot of the **KnxNetwork→KnxDevice→Comms Connections→Tunnel Conn**.



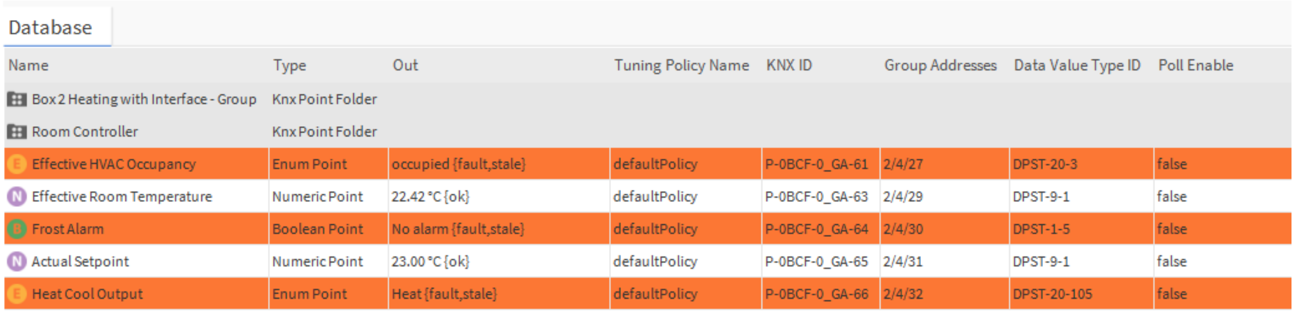
Property	8	wrongSeqNumberReaction	Wrong Seq Number Reaction	Frozen	h	knxnetIp:WrongSequenceNumberReactionEnum	
Property	9	includeInTrace	Include In Trace	Frozen	hN	baja:Boolean	
Property	10	individualAddress	Individual Address	Frozen	rt	knxnetIp:IndividualDeviceAddress	
Property	11	confirmationTimeout	Confirmation Timeout	Frozen	rh	baja:RelTime	min=1 second,max=1 minute
Property	12	commsCounters	Comms Counters	Frozen	hN	knxnetIp:ConnectionCommsCounters	
Property	13	requestAcknowledgements	Request Acknowledgements	Frozen	h	baja:Boolean	

Change the **`Request Acknowledgements`** property to `false`.



Channel Id	71	[0 - 255]
Channel Status	Operation successful.	
Allways Send Heart Beats	true	
Individual Address	15.15.255	
Request Acknowledgements	false	

Why do some points randomly change state between {stale} and/or {fault} and {ok}?

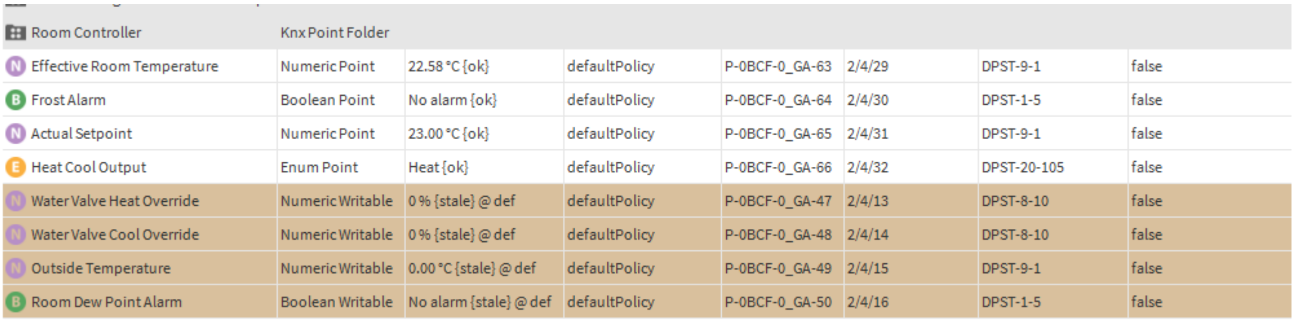


Name	Type	Out	Tuning Policy Name	KNX ID	Group Addresses	Data Value Type ID	Poll Enable
Box 2 Heating with Interface - Group	Knx Point Folder						
Room Controller	Knx Point Folder						
Effective HVAC Occupancy	Enum Point	occupied {fault,stale}	defaultPolicy	P-0BCF-0_GA-61	2/4/27	DPST-20-3	false
Effective Room Temperature	Numeric Point	22.42 °C {ok}	defaultPolicy	P-0BCF-0_GA-63	2/4/29	DPST-9-1	false
Frost Alarm	Boolean Point	No alarm {fault,stale}	defaultPolicy	P-0BCF-0_GA-64	2/4/30	DPST-1-5	false
Actual Setpoint	Numeric Point	23.00 °C {ok}	defaultPolicy	P-0BCF-0_GA-65	2/4/31	DPST-9-1	false
Heat Cool Output	Enum Point	Heat {fault,stale}	defaultPolicy	P-0BCF-0_GA-66	2/4/32	DPST-20-105	false

If, after appropriate time-outs, the KNXnet/IP driver fails to receive requested data from the KNX device, points go to a `{fault}` and/or `{stale}` state.

If the KNX device subsequently transmits data, for example a change of state or value, the KNXnet/IP driver receives this unsolicited message thereby causing a change of proxy point status.

Why, when discovered points are added to the Database, some are {stale} and some are not?



Room Controller	Knx Point Folder						
Effective Room Temperature	Numeric Point	22.58 °C {ok}	defaultPolicy	P-0BCF-0_GA-63	2/4/29	DPST-9-1	false
Frost Alarm	Boolean Point	No alarm {ok}	defaultPolicy	P-0BCF-0_GA-64	2/4/30	DPST-1-5	false
Actual Setpoint	Numeric Point	23.00 °C {ok}	defaultPolicy	P-0BCF-0_GA-65	2/4/31	DPST-9-1	false
Heat Cool Output	Enum Point	Heat {ok}	defaultPolicy	P-0BCF-0_GA-66	2/4/32	DPST-20-105	false
Water Valve Heat Override	Numeric Writable	0 % {stale} @ def	defaultPolicy	P-0BCF-0_GA-47	2/4/13	DPST-8-10	false
Water Valve Cool Override	Numeric Writable	0 % {stale} @ def	defaultPolicy	P-0BCF-0_GA-48	2/4/14	DPST-8-10	false
Outside Temperature	Numeric Writable	0.00 °C {stale} @ def	defaultPolicy	P-0BCF-0_GA-49	2/4/15	DPST-9-1	false
Room Dew Point Alarm	Boolean Writable	No alarm {stale} @ def	defaultPolicy	P-0BCF-0_GA-50	2/4/16	DPST-1-5	false

One of the many reasons why points can go to a stale state relates to the ETS project file.

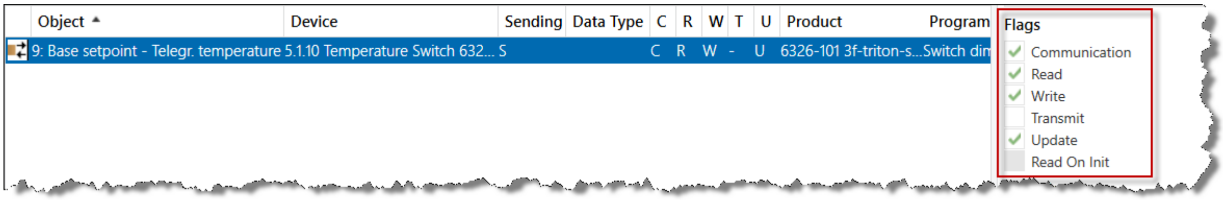
When points are added to the database from a ETS project file, their subscription properties are automatically configured based on the communications flags set in the ETS project. These are the flags for the KNX device's group objects, which are associated with the point's **`Group Address`**.

If a group address has at least one KNX device group object associated with it that has its `Read Communication Flag` property enabled, the **`Poll once on subscribed`**, **`Poll once on operational`** and **`Poll until answer after poll once`** properties are set to `true`. This triggers an immediate poll of the **`Group Address`** (because this view is itself subscribed to the point).

If a **`Group Address`** has no associated KNX device group objects that have their `Read Communication Flag` enabled, its subscription properties are all set to `false` by default. The value, however, may be updated (and become not stale) in the KNXnet/IP driver when any subsequent unsolicited messages regarding the **`Group Address`** are received from the KNX system.

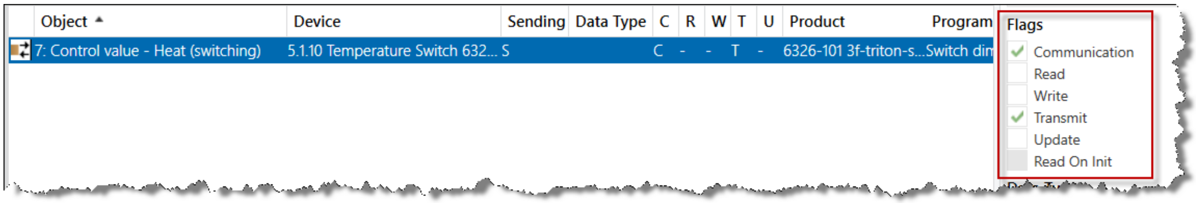
Here are the `Read Communication Flag` settings for some of the points in the example shown above:

NOTE: The Control Output (Control value) object has its `Transmit` flag set. This causes an unsolicited message transmission from the KNX device.



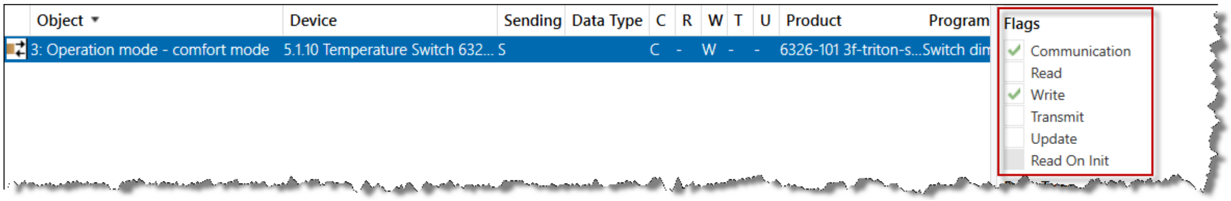
Object ▲	Device	Sending	Data Type	C	R	W	T	U	Product	Program
9: Base setpoint - Telegr. temperature	5.1.10 Temperature Switch 632...	S		C	R	W	-	U	6326-101 3f-triton-s...Switch di	

Flags
- ✓ Communication
- ✓ Read
- ✓ Write
- ☐ Transmit
- ✓ Update
- ☐ Read On Init



Object ▲	Device	Sending	Data Type	C	R	W	T	U	Product	Program
7: Control value - Heat (switching)	5.1.10 Temperature Switch 632...	S		C	-	-	T	-	6326-101 3f-triton-s...Switch di	

Flags
- ✓ Communication
- ☐ Read
- ☐ Write
- ✓ Transmit
- ☐ Update
- ☐ Read On Init



Object ▼	Device	Sending	Data Type	C	R	W	T	U	Product	Program
3: Operation mode - comfort mode	5.1.10 Temperature Switch 632...	S		C	-	W	-	-	6326-101 3f-triton-s...Switch di	

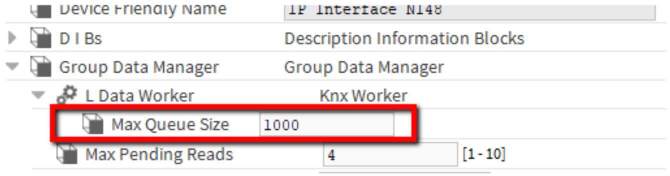
Flags
- ✓ Communication
- ☐ Read
- ✓ Write
- ☐ Transmit
- ☐ Update
- ☐ Read On Init

Please explain the Read fault: The ‘Read’ queue is full.

This fault occurs when an attempt is made to enqueue (add an item to a queue) a read request of a particular **`Group Address`**, where the queue is already full and the **`Group Address`** in question is not already in the queue. The queue in question is the Group Data Operation Queue, which holds a list of group data operations (`Group Address` reads or writes) waiting to be started as soon as the communications stack is able.

The Group Data Operation Queue has its size exposed under the hidden **L Data Worker** child of the **Group Data Manager**, immediately preceding the `Hop Count` property and can be seen in the example below. The default value of the Group Data Operation Queue is 50.

Subject to how many `Group Addresses` are being polled, how often, and the amount of available RAM, you could try increasing the value to 1000 to overcome the fault.



Device Friendly Name	IP Interface N148	
D I Bs	Description Information Blocks	
Group Data Manager	Group Data Manager	
L Data Worker	Knx Worker	
Max Queue Size	1000	
Max Pending Reads	4	[1 - 10]

Chapter 3 Components

Topics covered in this chapter

♦ knxnetIp-Connections
♦ knxnetIp-ConnectionsCommsCounters
♦ knxnetIp-KnxDevice
♦ knxnetIp-KnxDeviceFolder
♦ knxnetIp-EndPoint
♦ knxnetIp-EndPointCommsCounters
♦ knxnetIp-GroupDataManage
♦ knxnetIp-KnxInstallation
♦ knxnetIp-LocalInterface
♦ knxnetIp-LocalInterfaces
♦ knxnetIp-KnxNetwork
♦ knxnetIp-KnxPointDeviceExt
♦ knxnetIp-TunnelConnection
♦ knxnetIp-TunnelConnectionCommsCounters

Components include services, folders and other model building blocks associated with a module. You may drag them to a property or wire sheet from a palette.

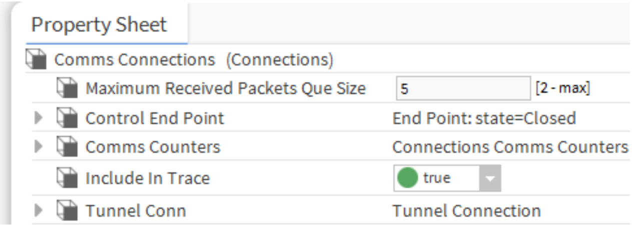
Descriptions included in the following topics appear as context-sensitive help topics when accessed by:

- Right-clicking on the object and selecting **Views→Guide Help**
- Clicking **Help→Guide On Target**

knxnetIp-Connections

These properties configure data communication options.

Figure 12 Comms Connections properties



Property Sheet

Comms Connections (Connections)

Maximum Received Packets Que Size	5 [2 - max]
Control End Point	End Point: state=Closed
Comms Counters	Connections Comms Counters
Include In Trace	true
Tunnel Conn	Tunnel Connection

To access these properties, expand **Config→Drivers→KnxNetwork→KnxDevice** in the station Nav tree, right-click the **Comms Connections** node, click **Views→AX Property Sheet**, and expand **Group Data Manager**.

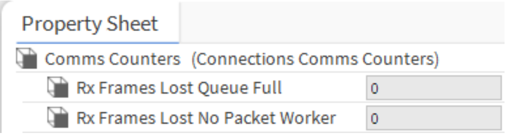
To configure the hidden **`L Data Worker`** and **`Hop Count`** properties you must first turn off the hidden config flag. To do so, right-click **Group Data Manager** in the Nav tree, click **Views→AX Slot Sheet**, right-click the hidden property row(s) in the table, click `Config Flags`, un-check the `Hidden` flag, and click **OK**.

Property	**Value**	**Description**
Maximum Received Packets Que Size	number (defaults to `5`)	Controls the size of the **Control End Point** receive queue: `2, 3 . . . max`. Because the **Control End Point** is only used to connect, maintain and disconnect the **Tunnel Connection**, it sees very little traffic. (only 2 or 3 messages to connect or disconnect and 1 or 2 messages per minute to maintain the connection). If this value were too small it would show up in the **Rx Frames Lost Queue Full** counter under the hidden **Comms Connections/ Comms Counters**.
Control End Point	additional properties End Point:	Holds information and configuration for the Control End Point: `state=Closed` or `state=Open`. Refer to knxnetIp-EndPoint, page 41.
Comms Counters (hidden)	Connections Comms Counters	Holds information on the Communications Counters. Refer to knxnetIp-ConnectionsCommsCounters, page 38.
Include In Trace (hidden)	`true` (default) or `false`	Controls whether or not this information is included in the station Spy Log.
Tunnel Conn	Tunnel Connection	Holds information and configuration for the Tunnel Connection. Refer to knxnetIp-TunnelConnection, page 53.

knxnetIp-ConnectionsCommsCounters

These properties provide communications statistics counters for the Comms Connections. By default, this is a hidden folder.

Figure 13 Connections Comms Counters properties



Property Sheet

Comms Counters (Connections Comms Counters)

Rx Frames Lost Queue Full 0

Rx Frames Lost No Packet Worker 0

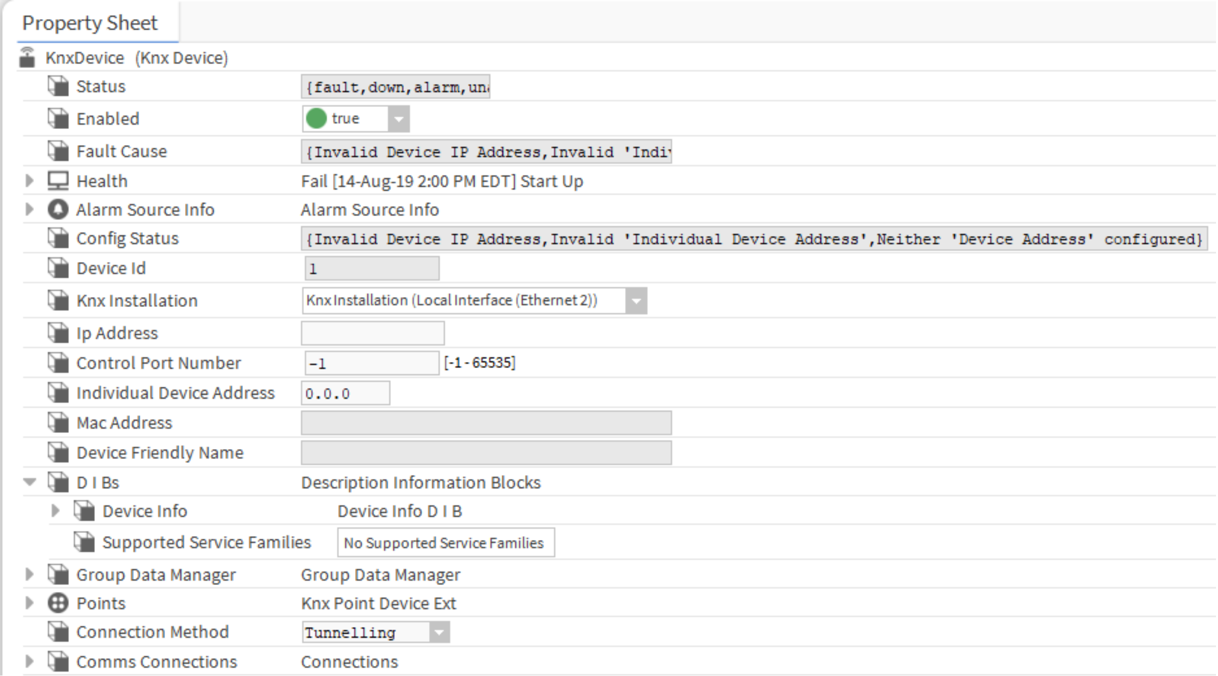
To access these hidden properties, you must first remove the config hidden flag, then expand **Config→Drivers→KnxNetwork→KnxDevice→Comms Connections** in the station Nav tree, and double-click the **Comms Counters**.

Property	**Value**	**Description**
Rx Frames Lost Queue Full	read-only	Reports the Rx frames lost when the queue is full : `0, 1, 2 . . . max`. The **Maximum Received Packets Que Size** controls the size of the **Control End Point's** receive queue size. It defaults of 5. If this value were too small, there would be an indication of lost frames here.
Rx Frames Lost No Packet Worker	read-only	Reports the Rx frames lost due to no packet worker: `0, 1, 2 . . . max`.

knxnetIp-KnxDevice

This component is the container for all KNXnet/IP proxy points.

Figure 14 Knx Device properties



Property Sheet

KnxDevice (Knx Device)

Status	{fault,down,alarm,un
Enabled	true
Fault Cause	{Invalid Device IP Address,Invalid 'Indi
Health	Fail [14-Aug-19 2:00 PM EDT] Start Up
Alarm Source Info	Alarm Source Info
Config Status	{Invalid Device IP Address,Invalid 'Individual Device Address',Neither 'Device Address' configured}
Device Id	1
Knx Installation	Knx Installation (Local Interface (Ethernet 2))
Ip Address	
Control Port Number	-1 [-1 - 65535]
Individual Device Address	0.0.0
Mac Address	
Device Friendly Name	
D I Bs	Description Information Blocks
Device Info	Device Info D I B
Supported Service Families	No Supported Service Families
Group Data Manager	Group Data Manager
Points	Knx Point Device Ext
Connection Method	Tunnelling
Comms Connections	Connections

To access these properties, expand **Config→Drivers→KnxDevice** in the station Nav tree, right-click the **KnxDevice** node, click **Views→AX Property Sheet**.

In addition to the standard properties (**`Status`**, **`Enabled`**, **`Fault Cause`**, **`Health`**, and **`Alarm Source Info`**), these properties are unique to the KNX Data Defs.

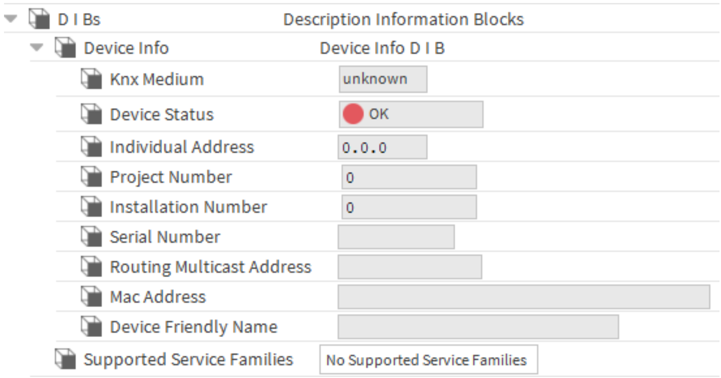
Property	Value	Description
Config Status	read-only	Displays the status of the configuration.
Device Id	read-only	Identifies the device.
Knx Installation	drop-down list (defaults to the current interface, including none selected)	Selects the multicast address group to which this device belongs.
Ip Address	`nnnn.nnnn.nnnn.nnnn`	Specifies the numerical label assigned to a device, which is connected to a computer network that uses the Internet Protocol for communication.
Control Port Number	`-1, 1, 2, ... 65535` (defaults to -1)	Identifies the port number on the controller or computer through which the software accesses a device.
Individual Device Address	`nnn.nnn.nnn`	Defines the unique number that identifies the current device object on the network.
Mac Address	read-only	Specifies the data link layer MAC address of the device.
Device Friendly Name	read-only	Displays an alternate name for the KNXnet/IP interface device.
D I Bs, Device Info	additional properties	DIBs are Description Information Blocks. These blocks configure devices. Refer to Device Info properties, page 40

Property	Value	Description
D I Bs, Supported Service Families	read-only	Indicates that the driver supports no service families.
Group Data Manager	Group Data Manager	Holds information and configuration for the Group Data Manager. Refer to knxnetIp-GroupDataManage, page 43.
Points	Knx Point Device Ext	Serves as a container for information and configuration properties for the KNX proxy points. Refer to knxnetIp-KnxPointDeviceExt, page 51
Connection Method	`Tunnelling` (default) or `Proxy Routing`	Displays the current mechanism for connecting the device to the station.
Comms Connections	Connections	Holds information and configuration of the Comms connections. Refer to knxnetIp-Connections, page 37.

Device Info properties

Description Information Blocks of the KNXnet/IP Interface device.

Figure 15 Device Info properties



D I Bs	Description Information Blocks
Device Info	Device Info D I B
Knx Medium	unknown
Device Status	● OK
Individual Address	0.0.0
Project Number	0
Installation Number	0
Serial Number	
Routing Multicast Address	
Mac Address	
Device Friendly Name	
Supported Service Families	No Supported Service Families

To access these properties, expand **Config→Drivers→KnxNetwork→KnxDevice** in the station Nav tree, right-click the **KnxDevice** node, click **Views→AX Property Sheet**, and expand **D I Bs→Device Info**.

Property	Value	Description
Knx Medium	read-only	Displays the Knx medium.
Device Status	read-only	Displays the current status of the device.
Individual Address	read-only	Displays the individual address for the device.
Project Number	read-only	Displays the ETS project identifier.
Installation Number	read-only	Displays the installation number.
Serial Number	read-only	Displays a serial number.
Routing Multicast Address	read-only	Displays the routing multicast address, in the form: `nnn.nnn.nnn.nnn`.

Property	Value	Description
Mac Address	read-only	Displays the of the KNXnet/IP interface device.
Device Friendly Name	read-only	Displays the alternate name for the KNXnet/IP interface device.

knxnetIp-KnxDeviceFolder

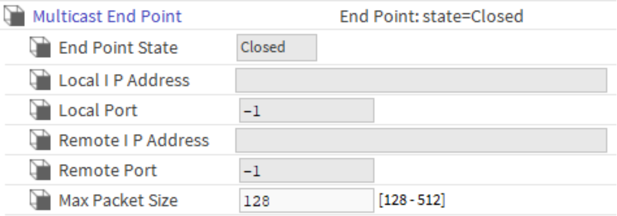
This component serves as a container for KNX devices.

Use this folder to organize devices in the station by geographic location or some other grouping.

knxnetIp-EndPoint

This component appears in the palette as the Multicast End Point. The same component appears under the KnxDevice, CommsConnection as a Control End Point and in a Tunnel Conn as a Data End Point. The properties for each end point are the same.

Figure 16 End Point properties



Multicast End Point End Point: state=Closed
End Point State Closed
Local I P Address
Local Port -1
Remote I P Address
Remote Port -1
Max Packet Size 128 [128 - 512]

Here is how to access the three versions of this component. All three instances of this component require you to expand **Config→Drivers→KnxNetwork** in the station Nav tree. From the **KnxNetwork** node, the procedure varies depending on the instance you are looking for.

Version to access	How to access
Multicast End Point	Right-click the **KnxNetwork** node, click **Views→AX Property Sheet**, expand **Local Interfaces→Local Interface→Knx Installation**, and **Multicast End Point** or double-click it. Double-clicking on **Multicast End Point** and clicking **Help→Guide On Target** opens this Multicast End Point help topic.
Control End Point	Expand **KnxDevice→Comms Connections** in the Nav tree, right-click the **Control End Point** node, and click **Views→AX Property Sheet** or double-click the **Control End Point** node.
Data End Point	Expand **KnxDevice→Comms Connections→Tunnel Conn** in the station Nav tree, right-click the **Data End Point** node, and click **Views→AX Property Sheet** or double-click the **Data End Point** node.

To configure the hidden **`Comms Counters`** and **`Include In Trace`** properties you must first turn off the hidden config flag. To do so, right-click end point in the Nav tree, click **Views→AX Slot Sheet**, right-click the hidden property row(s) in the table, click `Config Flags`, un-check the `Hidden` flag, and click **OK**.

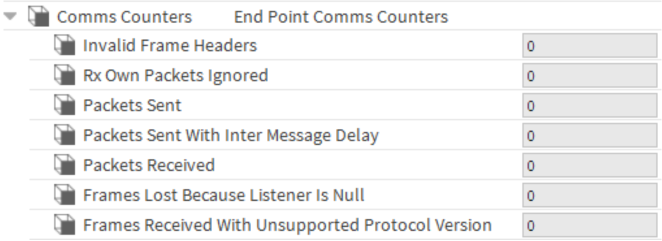
Property	Value	Description
End Point State	read-only	Reports the current state of the end point. `state=Closed` or `state=Open`
Local IP Address	read-only	Reports the local IP address in the format: `nnn.nnn.nnn.nnn`.

Property	**Value**	**Description**
Local Port	read-only	Reports the port number for the local port in the format: `nnnn`.
Remote IP Adddress	`Read-only`	Reports the IP address of the KNXnet/IP interface device in the format: `nnn.nnn.nnn.nnn`.
Remote Port	read-only	Reports the port number at KNXnet/IP interface device in the format: `nnnn`.
Max Packet Size	`numeric, 128 – 512` (Defaults to 128)	Defines the packet size.
Comms Counters (End Point Comms Counters) (hidden)	additional properties	Holds information on the Communications Counters. Refer to knxnetIp-EndPoint, page 41
Include In Trace (End Point Include In Trace) (hidden)	`true` (default) or `false`	Controls whether or not this information is included in the station Spy Log.

knxnetIp-EndPointCommsCounters

These properties provide communications statistics counters for the control end point. By default, this is a hidden folder.

Figure 17 End Point Comms Counters properties



Comms Counters	End Point Comms Counters
Invalid Frame Headers	0
Rx Own Packets Ignored	0
Packets Sent	0
Packets Sent With Inter Message Delay	0
Packets Received	0
Frames Lost Because Listener Is Null	0
Frames Received With Unsupported Protocol Version	0

The end point comms counters to view depends on your end point. There may be three instances in the station: Multicast End Point, Control End Point and Data End Point. To access these hidden properties, first change the hidden flag for **Comms Counters** properties, then expand **Config→Drivers→KnxNetwork→KnxDevice** in the station Nav tree, double-click one of the end point components (Multicast, Control or Data) and click the **Comms Counters** component.

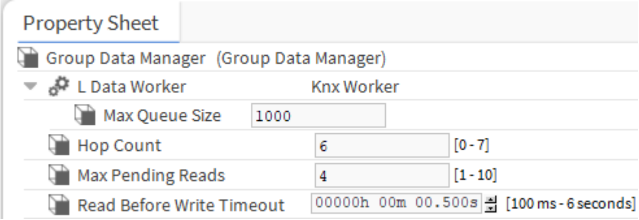
Property	**Value**	**Description**
Invalid Frame Headers	read-only	Reports invalid frame headers: `0, 1, 2 . . . max`.
Rx Own Packets Ignored	read-only	Reports ignored Rx Own packets: `0, 1, 2 . . . max`.
Packets Sent	read-only	Reports packets sent: `0, 1, 2 . . . max`.
Packets Sent With Inter Message Delay	read-only	Reports packets sent with an inter-message delay: `0, 1, 2 . . . max`.
Packets Received	read-only	Reports packets received: `0, 1, 2 . . . max`.

Property	Value	Description
Frames Lost Because Listener is Null	read-only	Reports frames lost because the listener contains null: `0, 1, 2 . . . max`.
Frames Received With Unsupported Protocol Version	read-only	Reports frames received with an unsupported version of the protocol: `0, 1, 2 . . . max`.

knxnetIp-GroupDataManage

These properties configure the **Group Data Manager** view.

Figure 18 Group Data Manager properties



Property Sheet

Group Data Manager (Group Data Manager)

L Data Worker Knx Worker

Max Queue Size 1000

Hop Count 6 [0 - 7]

Max Pending Reads 4 [1 - 10]

Read Before Write Timeout 00000h 00m 00.500s [100 ms - 6 seconds]

To access these properties, expand **Config→Drivers→KnxNetwork** in the station Nav tree, right-click the **KnxDevice** node, click **Views→AX Property Sheet**, and expand **Group Data Manager**.

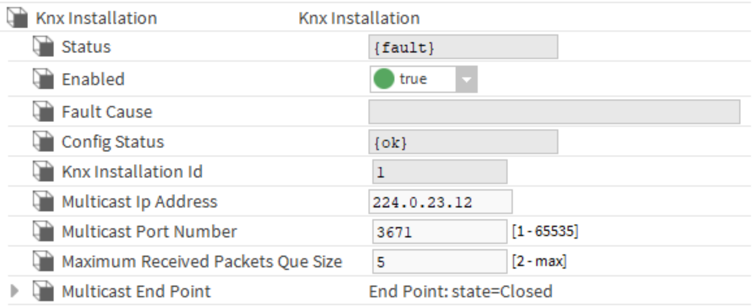
To configure the hidden **`L Data Worker`** and **`Hop Count`** properties you must first turn off the hidden config flag. To do so, right-click **Group Data Manager** in the Nav tree, click **Views→AX Slot Sheet**, right-click the hidden property row(s) in the table, click `Config Flags`, un-check the `Hidden` flag, and click **OK**.

Property	Value	Description
L Data Worker, Max Queue Size (hidden)	`1, 2, 3 . . . max` Defaults to 1000	Defines the size of the **Group Data Operation Queue** that holds a list of Group Address reads or writes waiting to be started as soon as the comms stack is able. **`L Data Worker`** stands for the Link Data Worker.
Hop Count (hidden)	`0, 1 , 2 . . . 7` (Defaults to 6)	Defines a counter for managing frames. Each communications frame contains a hop count. A frame's hop count is decremented by the KNXnet/IP Router devices to avoid looping messages. When it becomes zero, the frame is discarded from the KNX network.
Max Pending Reads	`1, 2, 3 . . . 10` (Defaults to 4)	Controls how many concurrently *Active* **Group Address Read Operations** are allowed. *Active* means that a particular operation has reached the head of the **Group Data Operation Queue** and the Comms.Stack has sent an `L_Data_req` message to the KNXnet/IP Interface devices and received an `L_Data_con` reply, but has not yet received a corresponding `L_Data_ind` message. This limit is intended to prevent the KNX twisted-pair line from being swamped with this driver's traffic although there is no clear guidance on this in the KNX specifications.
Read Before Write Timeout	`100ms . . . 6secs` (Defaults to 0.5 secs)	Defines the amount of time to wait before timing out on a read-before-write. Read Before Write Timeout.

knxnetIp-KnxInstallation

Holds information and configuration for the KNX Installation. In addition to the standard properties (Status, Enabled, and Fault Cause), these properties are unique to Knx Installation.

Figure 19 Knx Installation properties



Knx Installation	Knx Installation	
Status	{fault}	
Enabled	true	
Fault Cause		
Config Status	{ok}	
Knx Installation Id	1	
Multicast Ip Address	224.0.23.12	
Multicast Port Number	3671	[1 - 65535]
Maximum Received Packets Que Size	5	[2 - max]
Multicast End Point	End Point: state=Closed	

To access these properties, expand **Config→Drivers** in the station Nav tree, right-click the **KnxNetwork** node, click **Views→AX Property Sheet**, and expand **Local Interfaces→Local Interface→Knx Installation**.

In addition to the standard properties (Status, Enabled, and Fault Cause), this property sheet provides these properties.

Property	Value	Description
Config Status	read-only	Displays the current configuration state of the KNX driver.
Knx Installation Id	read-only	Displays the ID from the KNX Installation.
Multicast Ip Address	`nnn.nnn.nnn.nnn` (Defaults to 224.0.23.12)	Identifie the address used to broadcast a message to a group of destination computers simultaneously. (Do not confuse multicast with physical layer point-to-multipoint communication.)
Multicast Port Number	number (defaults to 3671)	Configures the port number to multi-cast.
Maximum Received Packets Que Size	number (Defaults to 5)	Configures the size of the Tunnel Connection's Data End Point receive queue.

knxnetIp-LocalInterface

KNXnet/IP interfaces are used for programming KNX systems from the Ethernet side. These properties hold information and configuration values for the local programming interface.

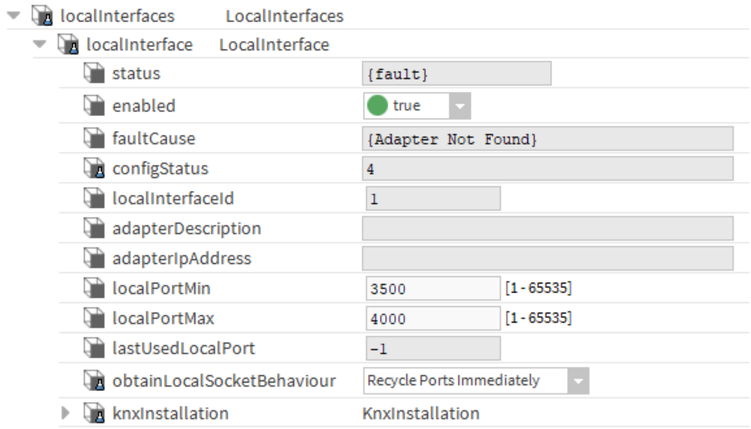


Figure 20 Local Interface properties

localInterfaces	LocalInterfaces	
localInterface	LocalInterface	
status	{fault}	
enabled	● true	
faultCause	{Adapter Not Found}	
configStatus	4	
localInterfaceId	1	
adapterDescription		
adapterIpAddress		
localPortMin	3500	[1 - 65535]
localPortMax	4000	[1 - 65535]
lastUsedLocalPort	-1	
obtainLocalSocketBehaviour	Recycle Ports Immediately	
knxInstallation	KnxInstallation	

To access these properties, expand **Config→Drivers** in the station Nav tree, right-click the **KnxNetwork** node, click **Views→AX Property Sheet**, and expand **Local Interfaces→Local Interface**.

In addition to the standard properties (**`Status`**, **`Enabled`**, and **`Fault Cause`**), these properties support the local interface function.

Property	**Value**	**Description**
Config Status	read-only	Displays the current configuration state of the local interface.
Local Interface Id	read-only	Identifies the interface by its number.
Adapter Id	drop-down list	Identifies the type of adapter among these options: Ehternet2, Ethernet, Local Area Connection* 9, Local Area Connetion* 11, Wi-Fi, Bluetooth Network Connection. Types that are not available include "[Disabled]" beside the option name.
Adapter Description	read-only	Describes the adapter.
Adapter IP Address	read-only	Reports the IP address of the adapter.
Local Port Min	`numeric, 1 — 65535` (Defaults to 3500)	Defines the lowest port number from which the driver dynamically assigns the next available number. Obtain Local Socket Behaviour enables dynamic port assignment.
Local Port Max	`numeric, 1 — 65535` (Defaults to 4000)	Defines the highest port number to which the driver dynamically assigns the next available number. Obtain Local Socket Behaviour enables this dynamic port assignment.
Last Used Local Port	read-only	Displays the port number assigned most recently.
Obtain Local Socket Behaviour	Drop-down list (defaults to Recycle Ports Immediately)	Selects how to assign ports. **`Recycle Ports`** Immediately causes the driver to assign the next port starting at Local Port Min. **`Cycle Through All Ports`** cycles through the range of ports.

knxnetIp-LocalInterfaces

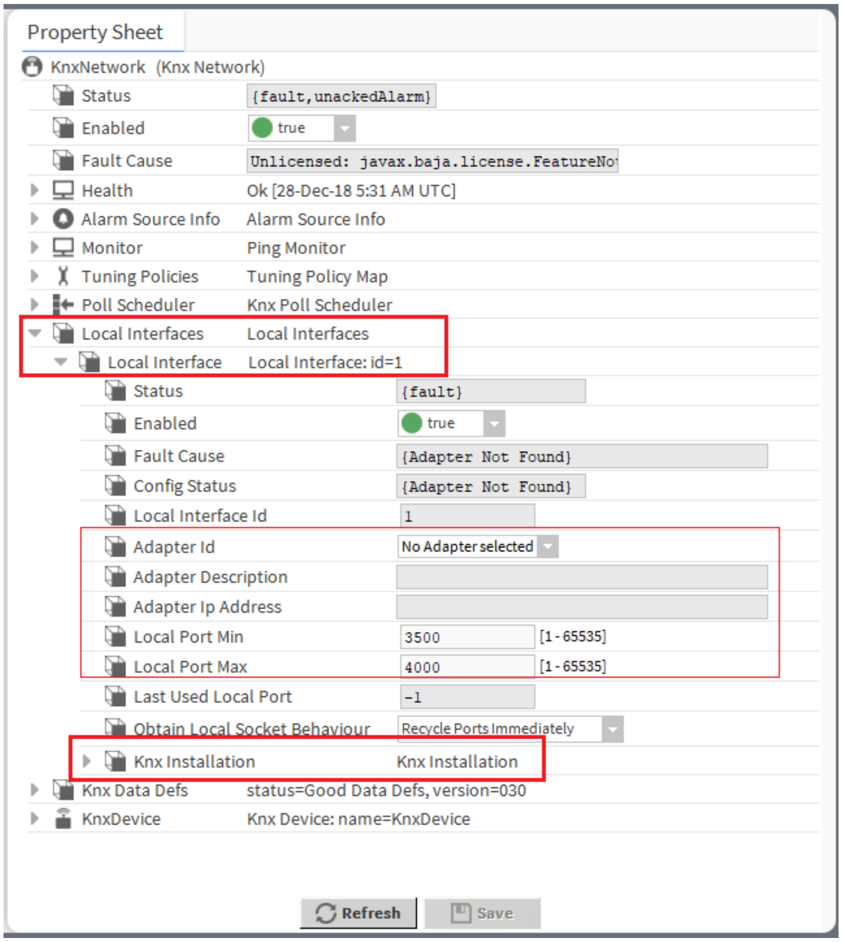
This component serves as a container under the KnxNetwork for local interfaces.

This is a default container under the KnxNetwork.

knxnetIp-KnxNetwork

This component is the base container for all KNXnet/IP devices and their child data objects (KNX proxy points).

Figure 21 KnxNetwork Property Sheet



KnxNetwork (Knx Network)

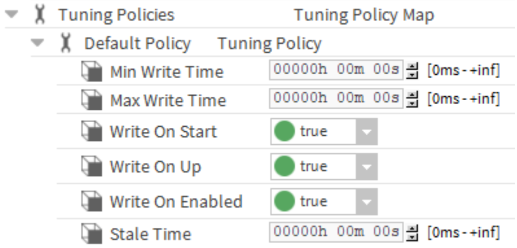
Status	{fault,unackedAlarm}
Enabled	true
Fault Cause	Unlicensed: javax.baja.license.FeatureNo
Health	Ok [28-Dec-18 5:31 AM UTC]
Alarm Source Info	Alarm Source Info
Monitor	Ping Monitor
Tuning Policies	Tuning Policy Map
Poll Scheduler	Knx Poll Scheduler
Local Interfaces	Local Interfaces
Local Interface	Local Interface: id=1
Status	{fault}
Enabled	true
Fault Cause	{Adapter Not Found}
Config Status	{Adapter Not Found}
Local Interface Id	1
Adapter Id	No Adapter selected
Adapter Description	
Adapter Ip Address	
Local Port Min	3500 [1 - 65535]
Local Port Max	4000 [1 - 65535]
Last Used Local Port	-1
Obtain Local Socket Behaviour	Recycle Ports Immediately
Knx Installation	Knx Installation
Knx Data Defs	status=Good Data Defs, version=030
KnxDevice	Knx Device: name=KnxDevice

Property	Value	Description
Status	read-only	Indicates the condition of the component at the last check. `{ok}` indicates that the component is licensed and polling successfully. `{down}` indicates that the last check was unsuccessful, perhaps because of an incorrect property, or possibly loss of network connection. `{disabled}` indicates that the **`Enable`** property is set to `false`. `{fault}` indicates another problem. Refer to **`Fault Cause`** for more information.
Enabled	`true` or `false` (defaults to `true`)	Activates and deactivates use of the component.
Fault Cause	read-only	Indicates the reason why a system object (network, device, component, extension, etc.) is not working properly (in fault). This property is empty unless a fault exists.
Health	read-only	Indicates the status of a system object (network, device or component) in the station. Includes a timestamp.
Alarm Source Info	additional properties	Contains a set of properties for configuring and routing alarms when this component is the alarm source.
Monitor	Ping Monitor	Configures a network's ping mechanism, which verifies network health. This includes verifying the health of all connected objects (typically, devices) by pinging each device at a repeated interval.
Tuning Policies	Tuning Policy Map	Refer to Tuning Policies/Default Policy, page 47.
Poll Scheduler	Knx Poll Scheduler	Refer to knxnetIp-KnxPollScheduler, page 49.
Local Interfaces	Local Interfaces	Refer to knxnetIp-LocalInterface, page 44.
Knx Data Defs	Additional properties	Refer to knxnetIp-KnxStationDataDefs, page 51.

Tuning Policies/Default Policy

The network Tuning Policies holds one or more collections of rules for evaluating both write requests (for example, to writable proxy points) as well as the acceptable freshness of read requests from polling.This is the default tuning policy which is always installed. More tuning policies may be created by copying and editing this default policy.

Figure 22 Tuning Policies, Default Policy properties



Tuning Policies Tuning Policy Map
Default Policy Tuning Policy
Min Write Time 00000h 00m 00s [0ms-+inf]
Max Write Time 00000h 00m 00s [0ms-+inf]
Write On Start true
Write On Up true
Write On Enabled true
Stale Time 00000h 00m 00s [0ms-+inf]

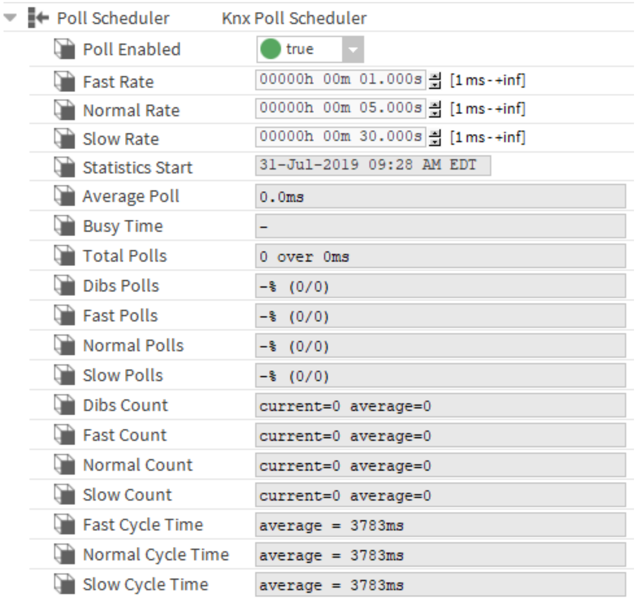
Property	Value	Description
Min Write Time	`hh-mm-ss, 0ms ... +inf` Defaults to 00–00–00	Specifies the minimum amount of time allowed between writes to writable proxy points, thus providing a method to throttle rapidly changing values so that only the last value is written. A value of zero (0) disables this rule causing all value changes to attempt to write.
Max Write Time	`hh-mm-ss, 0ms ... +inf` Defaults to 00–00–00	If nothing else triggers a write, this property specifies the maximum amount of time to wait before rewriting the value to the point. Any write action resets this timer. The default (zero) disables this rule resulting in no timed rewrites.
Write On Start	`true` (default) or `false`	Defines a writeable proxy point's behavior at station startup. `true` initiates a write when the station first reaches a steady state. `false` prevents a write when the station first reaches a steady state. Consider setting this to `false` for most tuning policies, except for those selectively assigned to more critical writable proxy points. This is particularly important for large networks with many writable proxy points. For example, a Network with 4,000 writable proxy points, if configured with only the default tuning policy (at default values), will upon station startup attempt to write to all 4,000 points, putting a significant load on the station. As a consequence, the driver (network) may generate write queue overflow exceptions.
Write On Up	`true` (default) or `false`	Defines a writeable proxy point's behavior when the point and its parent device transition from down to up. `true` initiates a write when the transition occurs. `false` prevents a write when the transition occurs.

Property	Value	Description
Write On Enabled	`true` (default) or `false`	Defines a writeable proxy point's behavior when its status transitions from disabled to enabled. `true` initiates a write when the transition occurs. `false` prevents a write when the transition occurs. The disabled-to-enabled status transition can be inherited globally by points if the parent device had been set to disabled—or network-wide if the driver network was set to disabled. Therefore, be aware that if left at `true`, all associated writable points receive a write upon either the device or network when it transitions from status disabled to enabled.
Stale Time	`hh-mm-ss, 0ms ... +inf` Defaults to 00–00–00	Defines the length of time after which the data for a point in the station database is considered to be too old to be meaningful (stale). A non-zero value causes the point to become stale (status stale) if the configured time elapses without a successful read, indicated by Read Status `{ok}`. Zero disables the stale timer. Points become stale immediately when unsubscribed. By default, the system indicates a stale proxy point status by a tan background color. In addition, stale status is considered invalud for any downstream-linked control logic.

knxnetIp-KnxPollScheduler

The network Poll Scheduler maintains a group of four rate buckets to service pollables, three of which correspond to configured poll rates (slow, normal and fast) and one (dibs stack) which is allocated for pollables that transition to a subscribed state.

Figure 23 Poll Scheduler properties



Poll Scheduler	Knx Poll Scheduler
Poll Enabled	● true
Fast Rate	00000h 00m 01.000s [1 ms - +inf]
Normal Rate	00000h 00m 05.000s [1 ms - +inf]
Slow Rate	00000h 00m 30.000s [1 ms - +inf]
Statistics Start	31-Jul-2019 09:28 AM EDT
Average Poll	0.0ms
Busy Time	-
Total Polls	0 over 0ms
Dibs Polls	-% (0/0)
Fast Polls	-% (0/0)
Normal Polls	-% (0/0)
Slow Polls	-% (0/0)
Dibs Count	current=0 average=0
Fast Count	current=0 average=0
Normal Count	current=0 average=0
Slow Count	current=0 average=0
Fast Cycle Time	average = 3783ms
Normal Cycle Time	average = 3783ms
Slow Cycle Time	average = 3783ms

To access these properties, expand **Config→Drivers** in the station Nav tree, right-click the **KnxNetwork** node, click **Views→AX Property Sheet**, and expand **Poll Scheduler** properties.

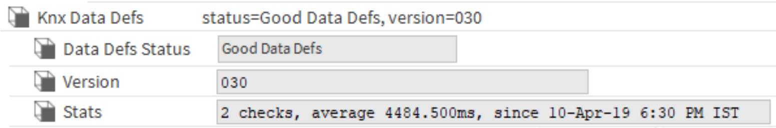
Property	**Value**	**Description**
Poll Enabled	`true` (default) or `false`	Enables or disables the Poll Scheduler.
Fast Rate	`hh-mm-ss, 1ms ... +inf` Defaults to 00–00–01	Sets the Fast Rate of Polling.
Normal Rate	`hh-mm-ss, 1ms ... +inf` Defaults to 00–00–05	Sets the Normal Rate of Polling.
Slow Rate	`hh-mm-ss, 1ms ... +inf` Defaults to 00–00–30	Sets the Slow Rate of Polling.
Statistics Start	read-only `date time`	Reports the last time statistics were reset.
Average Poll	read-only `time`	Reports the average time spent in each poll.
Busy Time	read-only `time`	Reports the percentage of the time that the station was busy doing polls.
Total Polls	read-only `Numeric` over `time`	Reports the total number of polls conducted and the time spent waiting for polls to execute.
Dibs Polls	read-only `Numeric` % (`Numeric`/ `Numeric`)	Reports the total number of polls made processing the dibs stack.
Fast Polls	read-only `Numeric` % (`Numeric`/ `Numeric`)	Reports the total number of polls made processing the fast queue.
Normal Polls	read-only `Numeric` % (`Numeric`/ `Numeric`)	Reports the total number of polls made processing the normal queue.
Slow Polls	read-only `Numeric` % (`Numeric`/ `Numeric`)	Reports the total number of polls made processing the slow queue.
Dibs Count	read-only current= `numeric` average= `numeric`	Reports the current and average number of components in the dibs stack.
Fast Count	read-only current= `numeric` average= `numeric`	Reports the current and average number of components in the fast queue.
Normal Count	read-only current= `numeric` average= `numeric`	Reports the current and average number of components in the normal queue.
Slow Count	read-only current= `numeric` average= `numeric`	Reports the current and average number of components in the slow queue.
Fast Cycle Time	read-only average = `time`	Reports the average cycle time for the fast queue.

Property	Value	Description
Normal Cycle Time	read-only average = `time`	Reports the average cycle time for the normal queue.
Slow Cycle Time	read-only average = `time`	Reports the average cycle time for the slow queue.

knxnetIp-KnxStationDataDefs

KNX Data Defs properties are part of the KNX Import Service. **Knx Data Defs** is a container for the specifications of the KNX Datapoint Types, which are imported from the `knx_extra.xml` file.

Figure 24 Knx Data Defs properties



Knx Data Defs status=Good Data Defs, version=030
Data Defs Status Good Data Defs
Version 030
Stats 2 checks, average 4484.500ms, since 10-Apr-19 6:30 PM IST

To access these properties, expand **Config→Drivers** in the station Nav tree, right-click the **KnxNetwork** node, click **Views→AX Property Sheet**, and expand **Knx Data Defs**.

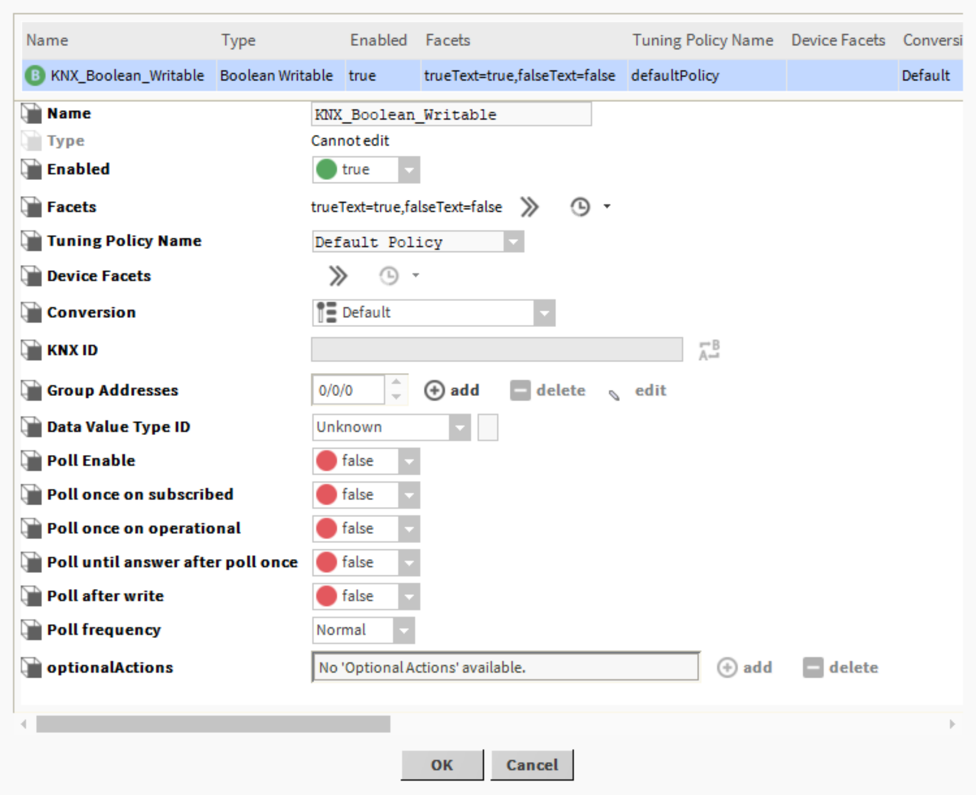
Property	Value	Description
Data Defs Status	read-only	Characterizes the state of the Data Defs.
Version	read-only	Reports the version of the Data Defs.
Stats	read-only	Provides additional information.

knxnetIp-KnxPointDeviceExt

All KNXnet/IP proxy extension types (**KnxBooleanProxyExt**, **KnxNumericProxyExt**, **KnxStringProxyExt** and **KnxEnumProxyExt**) share the same set of configuration properties. Any instance of any of the KNXnet/IP proxy extension types is a proxy for one or more group address in a KNX installation.

The KNXnet/IP proxy extension types take on the readable-writable personality of the control point they are attached to. For example, a **KnxNumericProxyExt**, when used as an extension on a numeric point has read-only functionality, but when used on as an extension on a numeric writable it can read and write attribute values.

Figure 25 Example of proxy extension properties



Name	Type	Enabled	Facets	Tuning Policy Name	Device Facets	Conversi
KNX_Boolean_Writable	Boolean Writable	true	trueText=true,falseText=false	defaultPolicy		Default

Name	KNX_Boolean_Writable
Type	Cannot edit
Enabled	true
Facets	trueText=true,falseText=false
Tuning Policy Name	Default Policy
Device Facets	
Conversion	Default
KNX ID	
Group Addresses	0/0/0
Data Value Type ID	Unknown
Poll Enable	false
Poll once on subscribed	false
Poll once on operational	false
Poll until answer after poll once	false
Poll after write	false
Poll frequency	Normal
optionalActions	No 'Optional Actions' available.

You access these properties by expanding **Config→Drivers→KnxNetwork→KnxDevice→Points** followed by double-clicking the point in the Nav tree or the row in the **KnxPointManager** view.

The KNXnet/IP proxy extension types are the point-level component in the Framework architecture. In addition to the standard properties (**`Status`**, **`Fault Cause`**, **`Enabled`**), these properties apply to the KNX Proxy extension. Other properties, **`Device Facets`**, **`Conversion`**, **`Tuning Policy Name`**, **`Read Value`**, and **`Write Value`** are defined in the Niagara Drivers Guide.

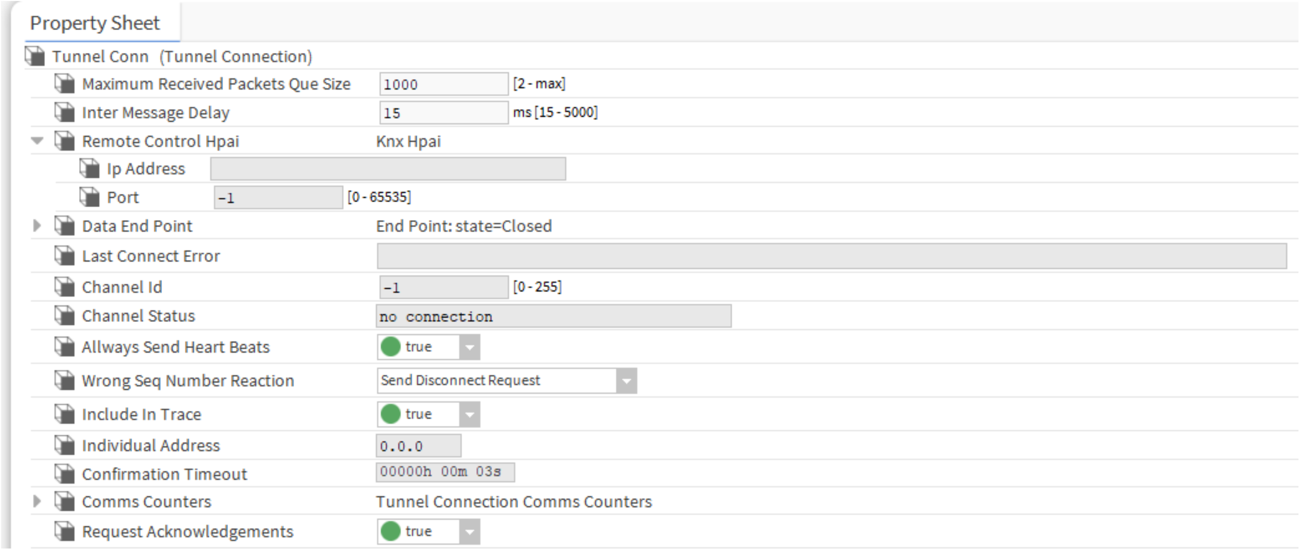
Type	Value	Description
KNX ID	read-only	Displays the KNX Installation ID.
Group Addresses	additional features	Defines a list of group addresses from which the value of this proxy extension can be updated. The first group address in the list is the primary group address and cannot be deleted, but can be edited. This proxy extension directs read and write requests to this address. The remaining group addresses update the proxy extension output value. This happens when the driver receives a message from any of these addresses.
Data Value Type ID	drop-down list (defaults to `Unknown`)	Defines the KNX data type for the address. It includes the size of the data value and its function. For example the data item may be a byte character or a one-byte percentage value. You typically use the data type as specified by the project file when you import the project, however, on occasion you may add points manually and will need to select the correct data type yourself.

Type	Value	Description
Poll Enable	`true` or `false`	Enables and disables adding and removing the group address (es) configured on this proxy extension to the Poll Scheduler. In KNX, a group address responds to a read request only if it is configured to do so. For group addresses, where there is no bus device configured to send the value in response to a read, it makes sense not to poll the group address. In other cases, it may desirable not to put additional traffic on the bus if that traffic is not absolutely necessary.
Poll Once On Subscribed	`true` or `false`	Forces a poll when a point enters a subscribed state, such as when a user views it on a point list. If enabled, the resulting poll occurs independently of any other poll setting (for instance, it occurs even if **Poll Enable** is `false` and if the **Poll Scheduler** rate is zero or disabled.) To modify this behaviour, configure the **Poll Until Answer Received On Poll Once** property.
Poll Once On Operational	`true` or `false`	Forces a poll when the point status changes from disabled to enabled, down to up, and fault to noFault. If enabled, the resulting poll is independent, and occurs independently of any other poll setting (for instance, it occurs even if **Poll Enable** is `false` and if the **Poll Scheduler** rate is zero or disabled.) To modify this behaviour, configure the **Poll Until Answer Received On Poll Once** property.
Poll Until Answer After Poll Once	`true` or `false`	Setting **Poll Once On Subscribed** or**Poll Once On Operational** to `true`, and this value to `true`, modifies the poll once behaviour to poll until the driver receives a valid value instead of polling once and discarding the value. This has the effect of subscribing the point to the **Poll Scheduler** until such time as the point receives data addressed to the first group address in the list provided the value is a valid value for this point type. When these conditions are satisfied, the point is unregistered from the poll scheduler.
Poll After Write	`true` or `false`	Enables and disables a poll for a value after a write. This property is independent of the **Poll Enable** property.
Poll Frequency	`Fast`, `Normal`, or `Slow`	Selects how often to poll the device. The **Poll Scheduler** defines these rates.

knxnetIp-TunnelConnection

These properties configure the tunnel connection.

Figure 26 Tunnel connection properties



Property Sheet

Tunnel Conn (Tunnel Connection)

Maximum Received Packets Que Size	1000	[2 - max]
Inter Message Delay	15	ms [15 - 5000]
Remote Control Hpai	Knx Hpai	
Ip Address		
Port	-1	[0 - 65535]
Data End Point	End Point: state=Closed	
Last Connect Error		
Channel Id	-1	[0 - 255]
Channel Status	no connection	
Allways Send Heart Beats	true	
Wrong Seq Number Reaction	Send Disconnect Request	
Include In Trace	true	
Individual Address	0.0.0	
Confirmation Timeout	00000h 00m 03s	
Comms Counters	Tunnel Connection Comms Counters	
Request Acknowledgements	true	

To access these properties, expand **Config→Drivers→KnxNetwork→KnxDevice→Comms Connections** in the station Nav tree, right-click the **Tunnel Connection** node, and click **Views→AX Property Sheet**.

To configure the hidden properties you must first turn off the hidden config flag. To do so, right-click **Tunnel Connection** in the Nav tree, click **Views→AX Slot Sheet**, right-click the hidden property row(s) in the table, click `Config Flags`, un-check the `Hidden` flag, and click **OK**.

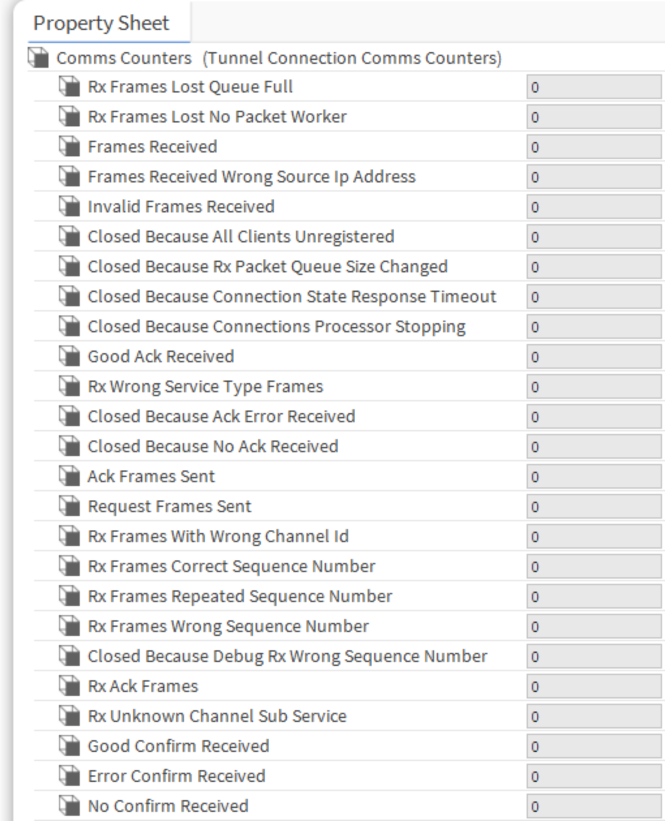
Property	Value	Description
Maximum Received Packets Que Size	number (defaults to `50`)	Controls the size of the Tunnel Connection's **Data End Point** receive queue: `2, 3 . . . max`. All **Group Address** messages pass through this queue and end point so it is much busier than the **Control End Point**. Because it handles more traffic, its default queue size of 50 is larger than that of the **Control End Point**.
Inter Message Delay	number (defaults to 15ms)	Paces the out-going data requests from the driver: `15ms . . . 5secs`. This can be used to reduce the rate of data requests in cases where a KNXnet/IP Interface device cannot cope with the traffic volume caused, possibly, by its implementation settings or by its operating speed. The default setting of minimum 15ms has proved successful with Siemens interface devices. Any problems arising from this being set too small would manifest as control points intermittently having a read fault: `Timed out waiting for L_Data_con`.
Remote Control Hpai, Ip Address	read-only	Reports the IP address in the format: `nnn.nnn.nnn.nnn`.
Remote Control Hpai, Port	read-only (defaults to –1 if no connection)	Reports the Port number in the format: `0, 1, . . . 65535`.
Data End Point	read-only	Reports the current state of the data end point: `state=Closed` or `state=Open`. Refer to .

Property	Value	Description
Last Connect Error	read-only	Reports any error returned at the last connection in the format: `string`.
Channel Id	read-only (defaults to –1 if no connection)	Reports the channel ID from `0 . . . 255`.
Channel Status	read-only	Reports the status of the channel in the format: `no connection` or `connection`.
Always Send Heart Beats	`true` (default) or `false`	Controls the reporting of heart beats.
Wrong Seq Number Reaction (hidden)	`Send Disconnect Request` (default) or `Just Ignore` or `Send NAK E_Sequence_Number`	Defines what reaction the driver should make to a Wrong Sequence Number.
Include In Trace (hidden)	`true` (default) or `false`	Controls if this information is included in the station Spy Log.
Individual Address	read-only	Reports the individual address in the format: `nn.nn.nn`.
Confirmation Timeout (hidden)	read-only	Confirms when the time out occurred in the format: `hh mm ss`.
Comms Counters (hidden)	Tunnel Connection Comms Counters	Holds information on the Communications Counters. Refer to knxnetIp-EndPointCommsCounters, page 42.
Request Acknowledgements (hidden)	`true` (default) or `false`	Request acknowledgements.

knxnetIp-TunnelConnectionCommsCounters

These properties provide communications statistics counters for the Tunnel Connection. By default, this is a hidden folder.

Figure 27 Tunnel Connection Comms Counters properties



Comms Counters (Tunnel Connection Comms Counters)

Rx Frames Lost Queue Full	0
Rx Frames Lost No Packet Worker	0
Frames Received	0
Frames Received Wrong Source Ip Address	0
Invalid Frames Received	0
Closed Because All Clients Unregistered	0
Closed Because Rx Packet Queue Size Changed	0
Closed Because Connection State Response Timeout	0
Closed Because Connections Processor Stopping	0
Good Ack Received	0
Rx Wrong Service Type Frames	0
Closed Because Ack Error Received	0
Closed Because No Ack Received	0
Ack Frames Sent	0
Request Frames Sent	0
Rx Frames With Wrong Channel Id	0
Rx Frames Correct Sequence Number	0
Rx Frames Repeated Sequence Number	0
Rx Frames Wrong Sequence Number	0
Closed Because Debug Rx Wrong Sequence Number	0
Rx Ack Frames	0
Rx Unknown Channel Sub Service	0
Good Confirm Received	0
Error Confirm Received	0
No Confirm Received	0

To access these hidden properties, first change the hidden flag for **Comms Counters**, then expand **Config→Drivers→KnxNetwork→KnxDevice→Comms Connections→Tunnel Conn** in the station Nav tree, right-click **Comms Counters** the node, click **Views→AX Property Sheet**.

Property	Value	Description
Rx Frames Lost Queue Full	read-only	Reports the Rx frames lost due to a full queue in the format: `0, 1, 2 . . . max`. You can monitor the success or otherwise of received data by inspecting this counter. Depending on the number and frequency of lost frames, you can try increasing the **Maximum Received Packets Que Size** in steps of 50, until no more frames are lost.
Rx Frames Lost No Packet Worker	read-only	Reports the Rx frames lost due to no packet worker in the format: `0, 1, 2 . . . max`.
Frames Received	read-only	Reports the frames received in the format: `0, 1, 2 . . . max`.
Frames Received Wrong Source IP Address	read-only	Reports the frames received with the wrong source IP address in the format: `0, 1, 2 . . . max`.
Invalid Frames Received	read-only	Reports the invalid frames received in the format: `0, 1, 2 . . . max`.
Closed Because All Clients Unregistered	read-only	Reports the number of times (in the format: `0, 1, 2 . . . max`) the transmission closed due to all clients being unregistered.

Property	Value	Description
Closed Because Rx Packet Queue Size Changed	read-only	Reports the number of times (in the format: `0, 1, 2 . . . max`) the transmission closed due to a change in the size of the Rx packet queue.
Closed Because Connection State Response Timeout	read-only	Reports the number of times (in the format: `0, 1, 2 . . . max`) the transmission closed due to a time out.
Closed Because Connections Processor Stopping	read-only	Reports the number of times (in the format: `0, 1, 2 . . . max`) the transmission closed due to the connections processor stopping.
Good Ack Received	read-only	Reports a good acknowledgment received in the format: `0, 1, 2 . . . max`.
Rx Wrong Service Type Frames	read-only	Reports the number of incorrect Rx service-type frames received in the format: `0, 1, 2 . . . max`.
Closed Because Ack Error Received	read-only	Reports the number of closures (in the format: `0, 1, 2 . . . max`) due to an acknowledgment error was received.
Closed Because No Ack Received	read-only	Reports the number of closures (in the format: `0, 1, 2 . . . max`) due to receiving no acknowledgment.
Ack Frames Sent	read-only	Reports the number of acknowledgment frames sent in the format: `0, 1, 2 . . . max`.
Request Frames Sent	read-only	Reports the number of request frames sent in the format: `0, 1, 2 . . . max`.
Rx Frames With Wrong Channel Id	read-only	Reports the number of Rx frames sent (in the format: `0, 1, 2 . . . max`) with the wrong channel Id.
Rx Frames Correct Sequence Number	read-only	Reports the number of Rx frames sent (in the format: `0, 1, 2 . . . max`) with the correct sequence number.
Rx Frames Repeated Sequence Number	read-only	Reports the number of Rx frames sent (in the format: `0, 1, 2 . . . max`) with a repeated sequence number.
Rx Frames Wrong Sequence Number	read-only	Reports the number of Rx frames sent (in the format: `0, 1, 2 . . . max`) with a wrong sequence number (read-only).
Closed Because Debug Rx Wrong Sequence Number	read-only	Reports the number of closures (in the format: `0, 1, 2 . . . max`) due to a debug Rx wrong sequence number.
Rx Ack Frames (hidden)	read-only	Reports the number of Rx acknowledgment frames (in the format: `0, 1, 2 . . . max`) sent.
Rx Unknown Channel Sub Service (hidden)	read-only	Reports the number (in the format: `0, 1, 2 . . . max`) of Rx unknown channel sub services.
Good Confirm Received (hidden)	read-only	Reports the number (in the format: `0, 1, 2 . . . max`) of good confirmation messages received.

Property	Value	Description
Error Confirm Received (hidden)	read-only	Reports the number (in the format: `0, 1, 2 ... max`) of error confirmation messages received.
No Confirm Received (hidden)	read-only	Reports the number (in the format: `0, 1, 2 ... max`) of no confirmation messages received.

Chapter 4 Windows

Topics covered in this chapter

- Discover Import Device
- Import an ETS project file
- New (Edit) KNX device

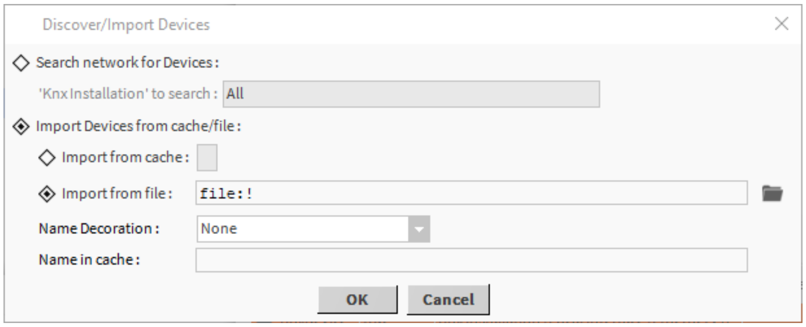
Windows create and edit database records or collect information when accessing a component. You access them by dragging a component from a palette to a nav tree node or by clicking a button.

Windows do not support **On View (F1)** and **Guide on Target** help. To learn about the information each contains, search the help system for key words.

Discover Import Device

This window filters the source of information used to add KNX devices to the station database.

Figure 28 Discover Import Device properties



Discover/Import Devices

◇ Search network for Devices :

'Knx Installation' to search : All

◈ Import Devices from cache/file :

◇ Import from cache :

◈ Import from file : `file:!`

Name Decoration : None

Name in cache :

You open this window when you double-click on the **KnxNetwork** node in the station Nav tree and click the **Discover** button.

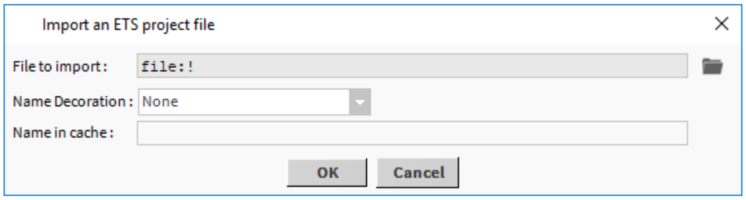
Property	Value	Description
Search network for Devices, 'Knx Installation to search'	radio button and drop-down list	Selects the group (installation) of KNX devices on the network to search: `Knx Installation (Local Interface...)`
Import Devices from cache/file, Import from cache	drop-down list	Selects a location in memory from which to acquire device information..
Import Devices from cache/file, Import from file	File Chooser	Selects a file, usually in the Supervisor PC, from which to acquire device information.

Property	Value	Description
Name Decoration	drop-down list	Optionally adds to the device name in cache the date and time the driver discovered the device. `Prepend File Date And Time` adds the data at the beginning of the record. `Append File Date And Time` adds the data at the end of the record.
Name in Cache	text	Defines a cache of devices discovered on previous searches. You might use this cache if a discovery is currently not possible.

Import an ETS project file

This window defines the ETS project file that contains the data definitions.

Figure 29 Import an ETS project file



Import an ETS project file

File to import: file:!

Name Decoration: None

Name in cache:

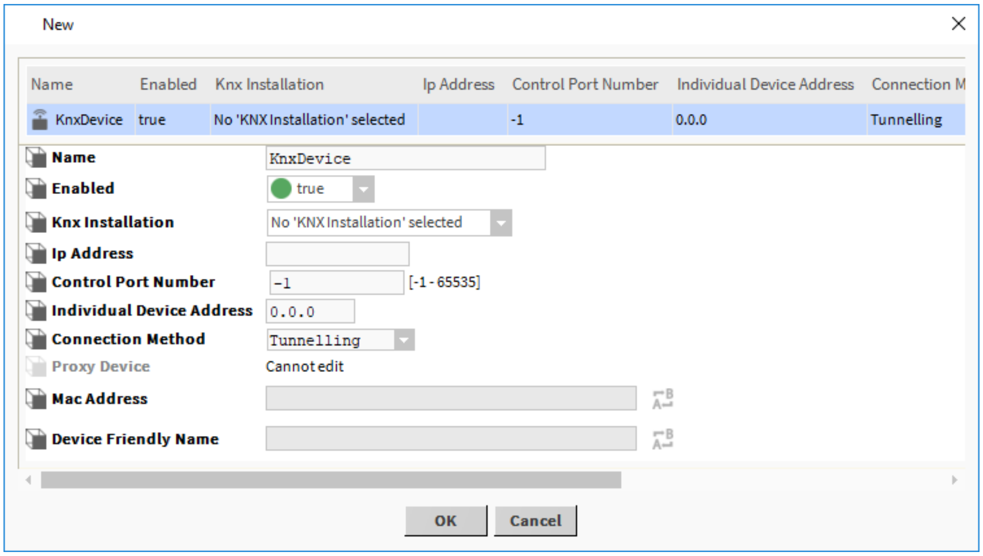
You open this window after clicking **Tools→KNX Import Service** by clicking on the **Imported Files**, slot followed by clicking the **Import** button.

Property	Value	Description
File to import	File Chooser	Identifies the location of the ETS project file on the Supervisor PC.
Name Decoration	drop-down list (defaults to `None`)	Optionally adds to the device name in cache the date and time the driver discovered the device. `Prepend File Date And Time` adds the data at the beginning of the record. `Append File Date And Time` adds the data at the end of the record.
Name in cache	text	Defines a cache of devices discovered on previous searches. You might use this cache if a discovery is currently not possible.

New (Edit) KNX device

This window manually configures device properties

Figure 30 New KNX Device properties



Name	Enabled	Knx Installation	Ip Address	Control Port Number	Individual Device Address	Connection M
KnxDevice	true	No 'KNX Installation' selected		-1	0.0.0	Tunnelling

Name	KnxDevice
Enabled	true
Knx Installation	No 'KNX Installation' selected
Ip Address	
Control Port Number	-1 [-1 - 65535]
Individual Device Address	0.0.0
Connection Method	Tunnelling
Proxy Device	Cannot edit
Mac Address	
Device Friendly Name	

You open this window when you double-click on the **KnxNetwork** node in the station Nav tree and click the **New** button.

Property	**Value**	**Description**
Name	text	
Knx Installation	drop-down list (defaults to the current interface, including none selected)	Selects the multicast address group to which this device belongs.
Ip Address	`nnnn.nnnn.nnnn.nnnn`	Specifies the numerical label assigned to a device, which is connected to a computer network that uses the Internet Protocol for communication.
Control Port Number	`-1, 1, 2, ... 65535` (defaults to -1)	Identifies the port number on the controller or computer through which the software accesses a device.
Individual Device Address	`nnn.nnn.nnn`	Defines the unique number that identifies the current device object on the network.
Connection Method	drop-down list (defaults to Tunnelling)	Defines how this KNX device routes messages to a KNX router.
Mac Address	read-only	Specifies the data link layer MAC address of the device.
Device Friendly Name	read-only	Displays an alternate name for the KNXnet/IP interface device.

Chapter 5 Plugins (views)

Topics covered in this chapter

♦ knxnetIp-KnxDeviceManager

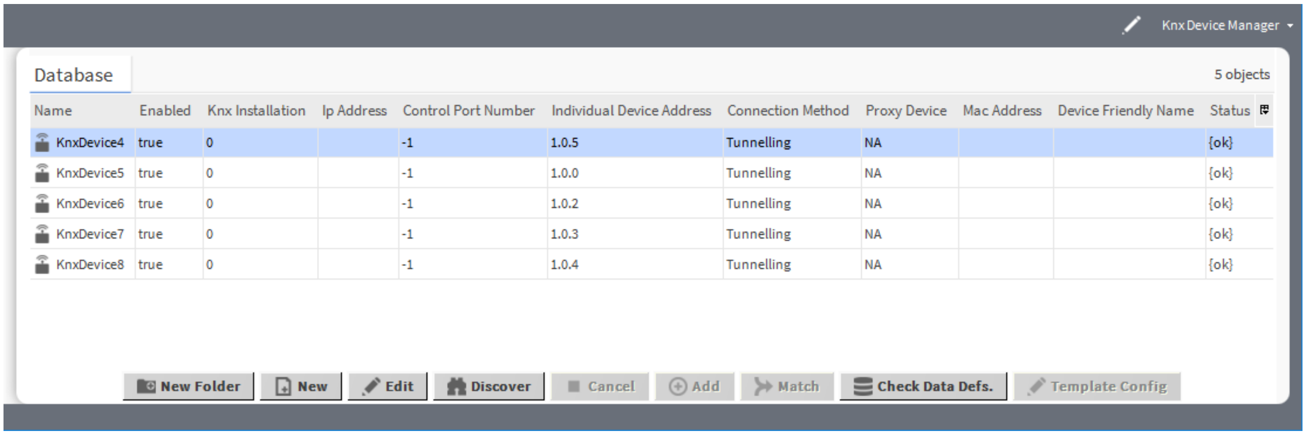
Plugins provide views of components and can be accessed in many ways. For example, double-click a component in the Nav tree to see its default view. In addition, you can right-click on a component and select from its **Views** menu.

For summary documentation on any view, select **Help→On View** (**F1**) from the menu or press **F1** while the view is open.

knxnetIp-KnxDeviceManager

This view lists the KNX devices on the network.

Figure 31 KNX Device Manager view



Name	Enabled	Knx Installation	Ip Address	Control Port Number	Individual Device Address	Connection Method	Proxy Device	Mac Address	Device Friendly Name	Status
KnxDevice4	true	0		-1	1.0.5	Tunnelling	NA			{ok}
KnxDevice5	true	0		-1	1.0.0	Tunnelling	NA			{ok}
KnxDevice6	true	0		-1	1.0.2	Tunnelling	NA			{ok}
KnxDevice7	true	0		-1	1.0.3	Tunnelling	NA			{ok}
KnxDevice8	true	0		-1	1.0.4	Tunnelling	NA			{ok}

You access this **Database** view by double-clicking the **Config→Drivers→KnxNetwork** node in the Nav tree.

Columns

Column name	Description
Connection Method	Indicates how this device routes messages to a KNX router.
Control Port Number	Identifies the port number on the controller or computer through which the software accesses a device.
Device Friendly Name	Displays an alternate name for the KNXnet/IP interface device.
Enabled	Reports if the function is turned on (true) or off (false).
Individual Device Address	Reports the unique number that identifies this device on the network.
Ip Address	Specifies the numerical label assigned to a device, which is connected to a computer network that uses the Internet Protocol for communication.
Knx Installation	Identifies the group of devices to which this device belongs.
Mac Address	Specifies the data link layer MAC address of the device.
Name	

Column name	Description
Path	Reports the location of the device in the station.
Proxy Device	If you are using proxy routing, this column displays the name of another KNX device that acts as the proxy for this device.
Status	Reports the current condition of the entity as of the last refresh: {alarm}, {disabled}, {down}, {fault}, {ok}, {stale}, {unackedAlarm}
Type	Identifies the type of KNX device.

Buttons

Button name	Description
New Folder	Creates a folder in the station for organizing devices.
New	Opens the New window with which to manually add a device to the database.
Edit	Opens an existing device record for updating.
Discover	Searches the network for KNX devices.
Cancel	Cancels the operation.
Add	Saves the discovered record in the station database.
Match	Associates a record in the database with the physical device.
Check Data Defs.	Updates the data definitions in the station.
TagIt	When tag mode is enabled for the device, this button opens a window for adding meta data tags to the selected device.
Template Config	Opens a window with template configuration properties.

Glossary

BatiBUS	This protocol was an open field bus for home appliances control and communication. After merging with two other protocols, it is part of the KNX standard.
BMS	Building Management System
EHS	The European Home Systems (EHS) protocol was aimed at home appliances control and communication. Now, after merging with two other protocols, it is part of the KNX standard
EIB	The European Installation Bus or Instabus is a decentralised open system to manage and control electrical devices within a facility. Now, after merging with two other protocols, it is part of the KNX standard
ETS	The Engineering Tool Software (ETS) is a PC software tool which enables the design, engineering and configuration of installations based on KNX certified products. The tool, which is manufacturer independent, enables a system integrator to combine products from different manufacturers into one solution
ETS project	This term describes an entity that consists of a group of KNX devices and the links between them. It may also contain KNX catalog data. The ETS software tool manages and maintains these projects.
KNX	KNX is a worldwide Standard for control in both commercial and residential buildings
KNX device	KNX devices are KNX system components that are connected together by a two-wire bus allowing them to exchange data. Sensors and actuators are examples of typical KNX devices.
KNX installation	A KNX installation comprises KNX devices, which are accessible through a KNX IP device.
KNXnet/IP driver	This is the Niagara KNXnet/IP driver that supports the standard KNX network protocol.
KNXnet/IP routing	This term refers to a multicast-based telegram, which allows a KNX IP router to perform the function of a line or area coupler.
KNXnet/IP Tunnelling	This term refers to the primary method of interfacing to a KNX system, which enables point-to-point communication (unicast) from a single external device to the KNX system. This is akin to using a USB or serial interface.
OSI	The Open Systems Interconnection model (OSI model) is a conceptual model that standardizes the communications functions of a telecommunication or computer system without regard to their underlying internal structure and technology. Its goal is the interoperability of diverse communication systems with standard protocols.

Index