

JENE-EG534-N4



niagara⁴

Trademarks

Onyx[®] and the Onyx logo are U.S. registered trademarks of Lynxspring, Inc.

Lynxspring[™] and the Lynxspring logo are U.S. registered trademarks of Lynxspring, Inc.

JENEsys[®] Edge[™] is a trademark of Lynxspring, Inc.

Other brands and product names mentioned in this manual may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.

Copyright Notice

Copyright © 2017, Lynxspring, Inc.

All rights reserved.

This document contains proprietary information which is protected by copyright. No part of this document may be copied, photocopied, reproduced, translated, or converted to any electronic or machine-readable form in whole or in part without prior written approval of Lynxspring, Inc.

Disclaimer

NO WARRANTY. This technical documentation is being delivered to you AS-IS, and Lynxspring makes no warranty as to its accuracy or use. Any use of the technical documentation or the information contained therein is at the risk of the user. Documentation may include technical or other inaccuracies or typographical errors.

Lynxspring reserves the right to make changes in this document without prior notice, and the reader should in all cases consult Lynxspring to determine whether any such changes have been made. The information in this publication does not represent a commitment on the part of Lynxspring.

Lynxspring shall not be liable for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

This guide contains links and references to third-party websites that are not under the control of Lynxspring, and Lynxspring is not responsible for the content of any reference material or linked websites. If you access a third party website mentioned in this guide, then you do so at your own risk. Lynxspring provides these links only as a convenience, and the inclusion of the link does not imply that Lynxspring endorses or accepts any responsibility for the content on those third-party sites.

Lynxspring, Inc.

1210 NE Windsor Drive

Lee's Summit, MO 64086

P 816.347.3500 F 816.437.0780

<http://www.lynxspring.com>

Table of Contents

Niagara 4 Documentation Addenda	4
JACE-8000 Startup Guide	4
Niagara 4 Platform Guide.....	5
Onyxx Network Driver Installation	5
Onyxx Network Quick Start	5
Configure the Onyxx Network	5
Add an onyxxNetwork.....	5
Discover and add Onyxx Devices	5
Manually adding Onyxx Devices.....	6
Add Onyxx Proxy points	6
Using online Discover to add Onyxx Proxy points	6
To discover I/O points.....	6
To add discovered I/O points as Onyxx Proxy points.....	6
Manually adding Onyxx Proxy points	7
Onyxx Network Concepts	7
About Onyxx Network Architecture	7
About the Onyxx Network	8
Onyxx Network status notes	8
Onyxx Network monitor notes	9
Onyxx Network tuning policy notes	9
Onyxx Network poll scheduler notes	9
Onyxx Network views.....	9
Onyxx Device Manager	9
Onyxx Board Manager usage notes	10
About the Onyxx Device	11
Onyxx Device properties	11
Onyxx Point Manager	12
Onyxx Driver Point Manager usage notes.....	12
About Onyxx Driver Proxy points.....	14
Onyxx Driver point types	15
Onyxx Driver Proxy Ext common properties	16
Conversion types in Onyxx Driver Proxy points.....	16
Scale and offset calculation (linear).....	18
Non-linear sensor support.....	19
Linear Calibration Ext	20
BooleanInputPoint	22
BooleanOutputWritable.....	23
CounterInputPoint.....	23
ResistiveInputPoint.....	23
ThermistorInputPoint.....	24
VoltageInputPoint.....	27
VoltageOutputWritable.....	28
Onyxx Network Driver Plugin Guides	28
Onyxx Network Driver Plugin Guides Summary	28
N Device Manager	28
Onyxx Driver Point Manager.....	29
JENEsys Edge Troubleshooting	30

The following are addenda to Niagara 4 documentation: docJace8000Startup and docPlatform

JACE-8000 Start Up Guide

Please note the following amendments to Niagara 4 documentation “docJace8000Startup” in regard to the operation of the JENEsys Edge 534 – N4. Referenced Niagara 4 documentation is included in the installation of ProBuilder™ or Niagara Framework® and is typically located in the docs directory of your JENEsys ProBuilder installation.

Chapter 1:

Pg. 7:

JENE-EG534-N4 controllers are pre-commissioned with Niagara 4.2.36 and therefore do not require commissioning to get started. Stations can be transferred onto them using JENEsys ProBuilder 4.2.36's Station Copier. Recommissioning and performing a factory reset requires the latest onyx modules which are included in JENEsys ProBuilder 4.2.36 -- available at: <http://resources.lynxspring.com/index.php/jenesys-home/jenesys-software/jenesys-probuilder>.

Pg. 7:

Converting a JENEsys Edge 534 – N4 to run AX is not available at this time.

Pg. 8:

WiFi is not available at this time.

Pg. 9 – IP address:

The JENEsys Edge 534 – N4 is shipped with a default address in the range of 192.168.1.2n, like the JACE-3E, JACE-6E, JACE-2/6 series.

Pg. 9:

The JENEsys Edge 534 – N4 does not use an SD card.

Chapter 2:

Pg. 11 – Provide power and connectivity:

Refer to the "JENEsys Edge 534 – N4 Wiring Guide".

Pg. 11 - Software version and Host Platform requirements:

Niagara 4.2 is the only supported version at this time.

Pg. 12 - Step 6, Note:

Serial Shell mode is not available at this time.

Pg. 12 - Connect to the JACE:

A Platform Connection is required to install Niagara 4.2 core software.

Chapter 3:

N4-to-AX conversion is not available at this time.

Chapter 4:

Pg. 27 - Configuring TCP/IP settings:

The JENEsys Edge 534 – N4 does not support WiFi at this time.

Pg. 28 - Step 4, Note:

The JENEsys Edge 534 – N4's Secondary Ethernet port is enabled by default.

Niagara 4 Platform Guide

Please note the following amendments to Niagara 4 documentation “docPlatform” in regards to the operation of the JENEsys Edge 534 – N4:

Chapter 1:

Pg. 29:

Returning a JENEsys Edge 534 – N4 to near factory defaults will be available soon.

Onyxx Network Driver Installation

To use the Lynxpring Onyxx Network driver, you need to have a JENEsys Edge device or Onyxx extender module (XM) that has onboard I/O hardware points.

From your PC, use the Niagara 4 Workbench N4.2 installed with the “installation tool” option (checkbox “This instance of Workbench will be used as an installation tool”). This option installs the needed distribution files (.dist files) for commissioning various models of remote platforms. The dist files are located under your Niagara 4 install folder in a “sw\dist” subfolder.

Apart from installing the N4.2 version of the Niagara 4 distribution file in the JENEsys Edge 534 – N4, make sure to install the **onyxxDriver** module (upgrade if an older revision). Additionally, the **kitlo**, **ndio** and **nrio** modules should install automatically with the **onyxxDriver**.

Note: The kitlo module provides a “Generic Tabular” conversion selection for non-linear sensor support, plus has several types of thermistor “curve files” in its xml folder.

Note: The Ndio and Nrio module provides the following conversions:

- *BShunt500OhmConversion*
- *BTabularThermistorConversion*
- *BThermistorType3Conversion*

Note: Basic procedures for using the onyxxDriver, including online discovery of I/O points (and their addition to your Niagara 4 station), is in the next section. See “Onyxx Network Quick Start” below.

Onyxx Network Quick Start

This section provides a collection of procedures to use the **onyxxDriver** to make Onyxx Proxy Points, the station interface to physical I/O points. As with other Niagara 4 drivers, you can do most configuration from special “manager” views and property sheets using Workbench.

Configure the Onyxx Network

To add and configure the onyxxNetwork, perform the following main tasks:

- **Add an onyxxNetwork**
- **Discover and add Onyxx devices**

Add an onyxxNetwork

Use the following procedure to add an onyxxNetwork under the station’s Drivers container.

To add an onyxxNetwork in the station:

- Step 1: From the station’s Driver Manager view, click the **New** button.
- Step 2: Select **onyxxNetwork** and click **OK**. Renaming the network is available if desired.
- Step 3: Click **OK** to add the onyxxNetwork to the station.

Discover and add Onyx Devices

In the Onyx driver architecture, file directories act as “device-level” components.

To discover Onyx device(s):

- Step 1: From the station's N Device Manager view, click the **Discover** button to launch an Onyx Device Discovery job.
- Step 2: Add the desired Onyx device(s) to the database.
- Step 3: Click to highlight the discovered Onyx Device(s) you wish to add, then click the **Add** button. The Add dialog appears, in which you can accept all defaults (**Figure 3-5**).
- Step 4: Click **OK** to add the Onyx Device(s) to your station.

Manually adding Onyx Devices

You can manually add Onyx Devices using the **New** button in the N Device Manager, or by dragging in from the Onyx palette. However, this method is generally not recommended as the path for the device will not be automatically populated.

Add Onyx Proxy points

As with device objects in other drivers, each Onyx Device has a Points extension that serves as the container for Proxy points. The default view for any Points extension is the Point Manager (and in this case, the **Onyx Driver Point Manager**). It is used to create Onyx Proxy points under any Onyx device.

This section provides quick start procedures for both tasks, as follows:

- **Using online Discover to add Onyx Proxy points**
- **Manually adding Onyx Proxy points**

Using online Discover to add Onyx Proxy points

This is the recommended way to accurately add Onyx Proxy points under an Onyx Device. Use the following procedures:

- **To discover I/O points**
- **To add discovered I/O points as Onyx Proxy points**
 - Note: If your onyxNetwork has multiple Onyx Devices, repeat both procedures (discover and add) for each Device, until you have all I/O points proxied in the station.*

To discover I/O points

Perform this task to discover I/O points.

- Step 1: From the N Device Manager view, double-click on the Points icon in the Exts column.
- Step 2: Click the **Discover** button to learn what I/O points are on the Onyx Device. When the discovery job completes, discovered I/O points are listed in the top pane of the view in the “Discovered” table. Each I/O point occupies one row.

To add discovered I/O points as Onyx Proxy points

Perform this task to add Onyx Proxy points.

To add discovered I/O points as Onyx Proxy points:

- Step 1: Select the I/O point or points in the Database pane of the Onyx Driver Point Manager.
- Step 2: You can map selected points in the station in different ways:
 - Drag from the Discovered pane to the Database pane.
 - Double-click an item in the Discovered pane.
 - Right-click, then select **add**.*Note: All the above will open up an Add dialog box.*
- Step 3: When the Add dialog appears, you typically edit a number of fields for each I/O point. For complete details, see “Onyx Driver Point Manager usage notes” on page 12. The following brief summaries explain Add dialog fields:
 - **Name** - This is the Onyx Proxy point name—you typically change this to describe the I/O purpose.

- **Type** - This is the Onyx “type,” which is selectable for any “universal input,” but fixed for any output. See “Universal Input selection notes” on page 13 and “Output type selection notes” on page 14.
 - Note: Unlike other entries in the Add dialog, you cannot edit the point’s Type later.*
 - **Address** - This is the learned I/O point’s hardware port address.
 - **Poll Frequency** - This specifies the point’s poll frequency group (default is Normal).
 - **Conversion** - This specifies the conversion type used between the Onyx Proxy extension’s “Device Facets” and the parent point’s facets. See “Conversion types in Onyx Driver Proxy points” on page 16.
 - **Facets** - These are the Onyx Proxy point’s facets, for how the value should be displayed in Niagara 4.
- Step 4: When you have Onyx Proxy point(s) configured properly for your usage, click **OK**. The Proxy points will be added to the station, and appear in the Database pane. For more details, see “About Onyx Driver Proxy points” on page 14.

Note: In cases where you notice a fault status for input points you have edited with “non-default” facets, it is likely related to a mismatch of the “Units” in the point’s automatically-appended Linear Calibration Ext. To clear this fault status, expand the point’s Linear Calibration Ext and edit Units to match the units in the point’s facets. For more information, see “Linear Calibration Ext” on page 20.

Manually adding Onyx Proxy points

You can manually add Onyx Proxy points, using the “New” button in the Onyx Point Manager, or by dragging from the Onyx palette. However, this method is generally not recommended as the path for the point will not be automatically populated.

Onyx Network Concepts

This section provides conceptual details on the onyxNetwork driver and its components, including views. The onyxNetwork Driver components are the station’s interface to physical I/O points on JENEsys Edge 534 – N4 controllers and attachable I/O extender modules.

About Onyx Network Architecture

Essentially, The onyxNetwork is similar in functionality to the NdioNetwork and uses the standard Niagara 4 network architecture. For example, real-time data is modeled using Onyx Proxy points, which reside under an Onyx “device,” which in turn resides under an Onyx network container in the station’s Drivers Container (Drivers in Nav window). Hierarchically, the component architecture is: network, device, point extension, points.

Figure 3-1 onyxNetwork driver architecture

Name	Type	Port Address	Poll Frequency
Heating Core	BooleanOutputWritable	1	Normal
Compressor	BooleanOutputWritable	2	Normal
Fan	BooleanOutputWritable	3	Normal
Return Damper	VoltageOutputWritable	1	Normal
Output Damper	VoltageOutputWritable	2	Normal
Thermostat	ThermistorInputPoint	1	Normal
universalInputPoint	type.OnyxBooleanInputProxyExtPoint	14	Normal
universalInputPoint1	VoltageInputPoint	3	Normal
universalInputPoint2	CurrentInputPoint	5	Normal
universalInputPoint3	VoltageInputPoint	2	Normal
universalInputPoint4	CounterInputPoint	4	Normal
universalInputPoint5	ResistiveInputPoint	6	Normal
universalInputPoint6	ThermistorInputPoint	7	Normal

Similar to **Ndio**, “Points” is the only extension under an Onyx “device”, and the purpose of the onyxDriver is to configure (and Proxy) the actual I/O hardware points.

Note: You use “Manager” views of Onyx container components to add all Onyx components to your station, including Proxy points. In these views, the driver provides online “discovery” of available devices and points and will automatically populate path names for devices and points.

For more details see:

- About the Onyx Network
- Onyx Device Manager
- About the Onyx Device
- Onyx Driver Point Manager
- About Onyx Driver Proxy points

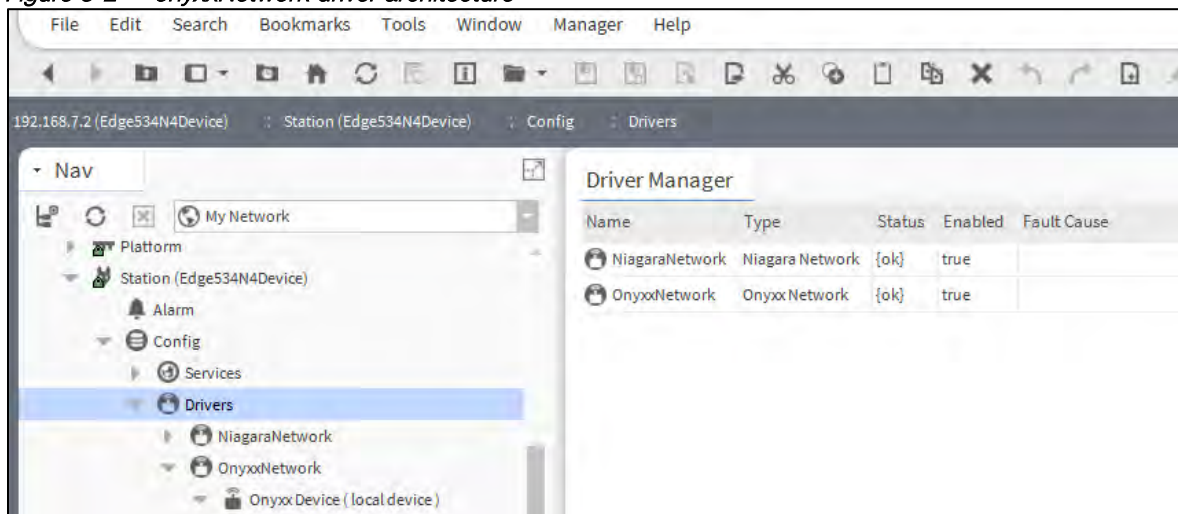
About the Onyx Network

The onyxNetwork is the top-level container component for “everything Onyx” in a station.

Note: Only one onyxNetwork component is valid in a station regardless of how many Onyx XM 34IOs or are present.

The onyxNetwork should reside in the station’s Drivers Container (“Drivers” in Nav window). The simplest way to add an onyxNetwork is from the Drivers Device Manager view, using the “New” command. Or, you can simply copy the onyxNetwork from the onyxDriver palette into Drivers container.

Figure 3-2 onyxNetwork driver architecture



Note: The JENEsys Edge 534 – N4 must have the Onyx modules installed.

The onyxNetwork component has the typical collection of slots and properties as most other network components. The following sections explain further:

- Onyx Network status notes
- Onyx Network monitor notes
- Onyx Network tuning policy notes
- Onyx Network poll scheduler notes
- Onyx Network views

Onyx Network status notes

Unlike most “fieldbus” drivers such as BACnet and Lonworks, the status of an onyxNetwork can be “down,” in addition to the normal “ok” or less typical “fault”. A down status might occur in the case of external I/O where the physical connection was removed (no communications are possible).

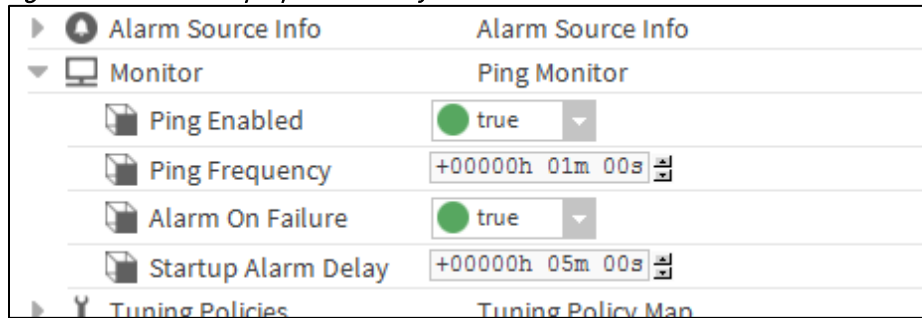
The Health slot contains historical timestamp properties that record the last network status transitions from “ok” to any other status. The “Fault Cause” property further explains any fault status.

Note: As in other driver networks, the onyxxNetwork has an available “Alarm Source Info” container slot you can use to differentiate onyxxNetwork alarms from other component alarms in the station. See “About network Alarm Source Info” in the Drivers Guide for more details.

Onyxx Network monitor notes

The Onyxx network’s monitor routine contains the basic network monitoring properties. For general information, see “About Monitor” in the Drivers Guide.

Figure 3-3 Monitor properties of onyxxNetwork



Onyxx Network tuning policy notes

The onyxxNetwork has the typical network-level Tuning Policy Map slot with a single default Tuning Policy. More information for this can be found in the “About Tuning Policies” in the Drivers Guide. However, given the small number of total Onyxx Proxy points supported under an onyxxNetwork, the importance of creating additional tuning policies (or editing the default tuning policy) is generally lower relative to other driver networks.

Onyxx Network poll scheduler notes

The onyxxNetwork has the typical network-level Poll Scheduler slot with standard collection of properties, as described in “About poll components” in the Drivers Guide. The following notes apply to the Poll Scheduler of an onyxxNetwork:

- By default, discovered/added Onyxx Proxy points have a Poll Frequency setting of “Normal” at the default Normal Rate of 5 seconds. Given the small number of total Onyxx Proxy points supported under an onyxxNetwork, you can safely assign them to the “Fast” rate (default 1 second).
- The Poll Frequency can be sped up or slowed down for Fast, Normal, and Slow as desired by the user.

Onyxx Network views

The onyxxNetwork’s default view is the N Device Manager, this is equivalent to the Device Manager in most other drivers. You use this view to discover and add Onyxx Device components to the station. For details, see “Onyxx Device Manager” below. Other standard views are also available on the onyxxNetwork.

Onyxx Device Manager

As the default onyxxNetwork view, the N Driver Device Manager can be used to discover and add one or more “device-level” Onyxx Device components to the station’s database (**Figure 3-4**). For general information, refer to “About the Device Manager” in the Drivers Guide.

Figure 3-4 Adding Onyx Device discovered using N Device Manager

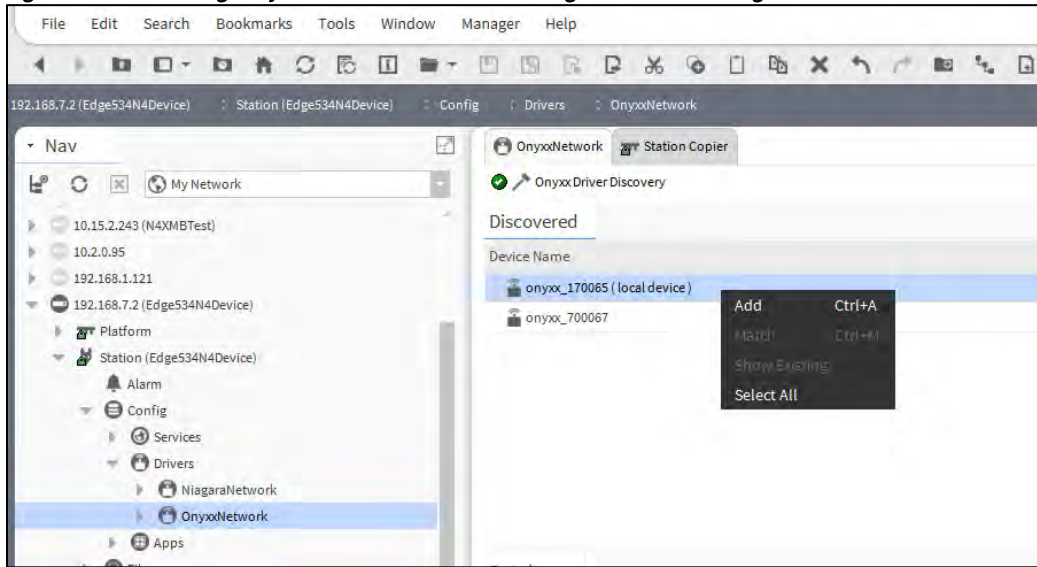
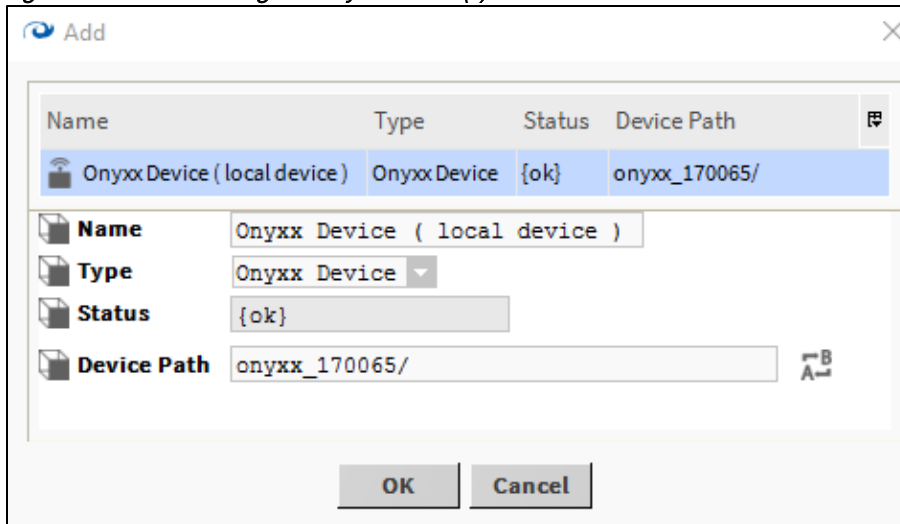


Figure 3-5 Add dialog for Onyx Device(s)



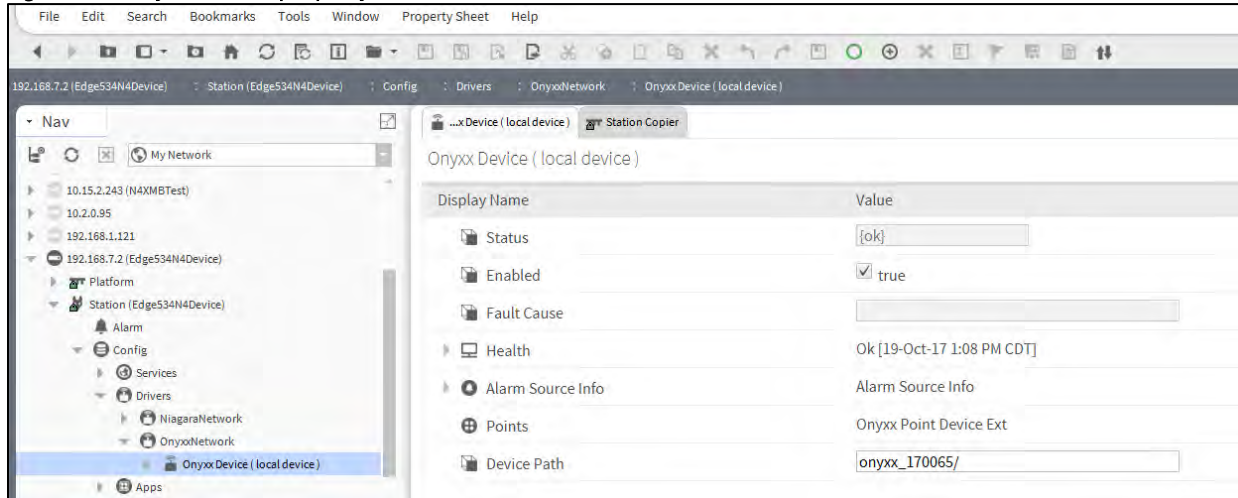
Onyx Board Manager usage notes

Before using discover, ensure that any external I/O modules are correctly installed and cabled to the host JENEsys Edge 534 – N4 controller. Additionally, you do not usually change any defaults except for “Name” in the Add dialog (Figure 3-5).

About the Onyx Device

The Onyx Device represents an I/O-34 board and can be extended by attaching Onyx XM modules. This JENEsys Edge 534 – N4 and its Onyx XM modules hold the processors that allow for servicing hardware I/O points.

Figure 3-6 Onyx Device property sheet



Under an onyxNetwork, each Onyx Device must have a unique “Device Path”. Apart from the “Enabled” property, this is the only writable property of an Onyx Device. All other Onyx properties are read-only types (Figure 3-6), reflecting the status and I/O capabilities.

Note: The N Device Manager view of the onyxNetwork provides online discover and add commands to ensure all available Onyx Devices can be created and configured properly.

Each Onyx Device has a single important device extension named Points. This is the container for all of a device’s Proxy points. Each Proxy point represents a specific I/O terminal address on the device (JENEsys Edge 534 – N4 or Onyx XM 34IO), serviced by that board’s I/O processor(s). For more details, see “Onyx Point Manager” on page 12.

Onyx Device properties

Onyx Device properties can be categorized into two groups:

- Device status properties
- Config properties

Note: As in other driver networks, the Onyx Device has an available “Alarm Source Info” container slot you can use to differentiate Onyx Device alarms from other component alarms in the station.

Device status properties

Onyx Devices have typical device-level status properties:

- **Status** - Status of the onyxNetwork communications to this Onyx Device; possible status flags include:
 - ❖ Ok - Normal communications, no other status flags
 - ❖ Disabled - Enabled property is set to false, either directly or in onyxNetwork and while status is disabled, all child onyxPoints have disabled status; Onyx Device polling is suspended
 - ❖ Fault - Typically configuration error, e.g. Onyx Device (duplicate or invalid IO Port number)
 - ❖ Down - Error communicating to one or more Onyx I/O processors on Onyx XM 34IO
- **Enabled** - Either true (default) or false; can be set directly or in the parent onyxNetwork; see Status disabled description above
- **Fault Cause** - If status has fault, describes the cause
- **Health** - Contains properties, including timestamps of last “ok” time and last “fail” time, and a string property describing the last fail cause

Config properties

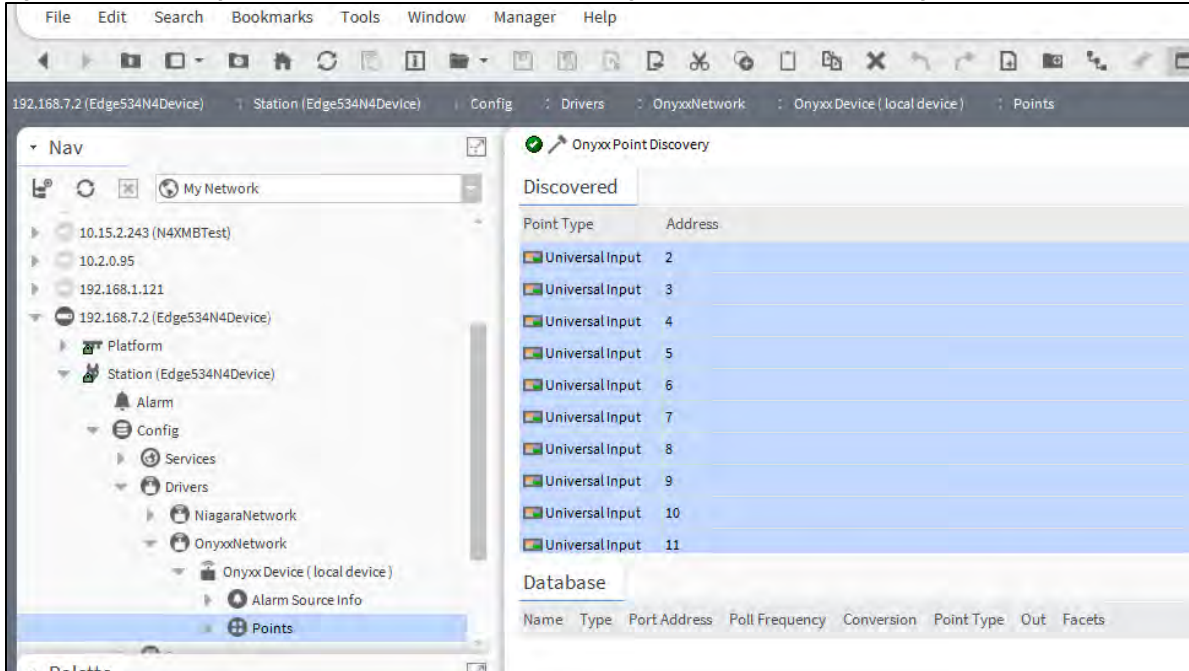
In addition to common device-level Device status properties, Onyx Device keeps track of its Directory Path for communication with the I/O ports.

- **Device Path** -- Set automatically if the device was discovered and added; a sub-portion of the full directory path

Onyx Point Manager

As the default view for the Points extension under an Onyx Device, the Onyx Driver Point Manager provides a discovery ability to find available I/O terminal “points” and add corresponding Onyx Proxy points to the station database.

Figure 3-7 Adding Onyx Proxy points discovered using Onyx Driver Point Manager



Typically you add a Proxy point for each discovered I/O terminal. For more details, see “Onyx Board Manager Usage notes” on page 10. For procedures, see “Add Onyx Proxy points” on page 6.

Onyx Driver Point Manager usage notes

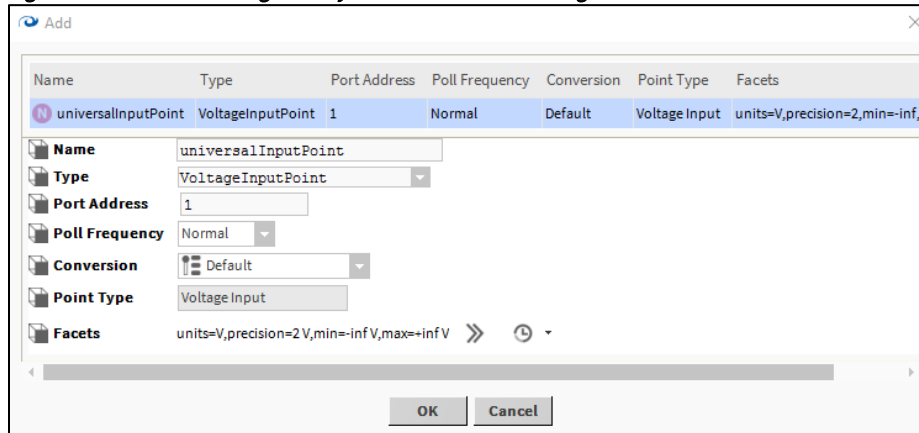
The following notes apply when using the Onyx Driver Point Manager:

- **Add and Edit dialog fields**
- **Universal Input type selection notes**
- **Output type selection notes**

Add and Edit dialog fields

When adding Onyx Proxy points for discovered I/O points, the following items are available in the Add dialog (Figure 3-8).

Figure 3-8 Add dialog in Onyx Driver Point Manager



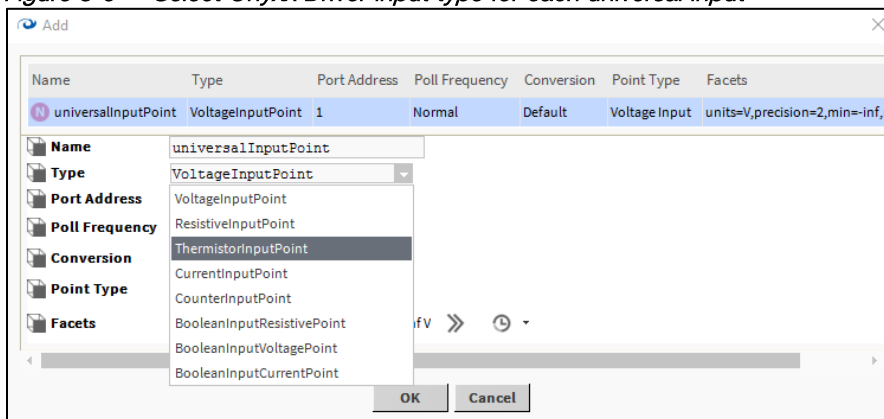
- **Name** - Name of the Onyx Proxy point (equivalent to right-click **Rename**, can be edited anytime); typically, you edit name from the default name to reflect the actual purpose of the I/O point, such as “Room101T,” “AHU-1_FanStatus,” and so forth
Note: The number appended onto a default name has no association to the I/O point address.
- **Type** - The type of Onyx Proxy point to create (not editable after adding, see “Universal Input selection notes” below)
- **Port Address** - The I/O terminal address within that general IO type; recommended to be left at discovered default
- **Poll Frequency** - The assigned poll frequency group, as configured in the Onyx Network’s poll scheduler; see “Onyx Network poll scheduler notes” on page 9
- **Conversion** - The conversion type used between units in the Proxy extension’s “Device Facets” and the units in the parent point’s Facets; see “Onyx Driver Proxy Ext common properties” on page 16 for more details
- **Point Type** - Closely related to the “Type” field; specifies if point will be of Current, Resistance, or Voltage point
- **Facets** - Facets for the Onyx Proxy point and equivalent to accessing facets through the point’s property sheet; these are automatically set during creation

Universal Input selection notes

All JENEsys Edge 534 – N4 controllers and external I/O modules provide 16 “universal inputs” (UI) terminals. On the device, these terminals are labeled UI1 to UI6.

Do this when adding by selecting the needed input type in the Add dialog, as shown in **Figure 3-9**.

Figure 3-9 Select Onyx Driver input type for each universal input



For most I/O platforms, any of the following are valid universal input choices:

- **VoltageInputPoint** - NumericPoint that reads a 0-to-10 Vdc input signal and produces either a voltage value or linear scaled output value.
- **ResistiveInputPoint** - NumericPoint that reads a resistive signal within a 0-to-100 K ohm range and produces either an ohms value or linear scaled output value.
- **ThermistorInputPoint** - NumericPoint that reads a Thermistor temperature sensor (Type 3, or other) signal and produces a scaled output value.
- **CurrentInputPoint** - NumericPoint that reads 4 mA to 20 mA and produces an amperage value.
- **CounterInputPoint** - NumericPoint that counts the number of contact closures at the input, and also calculates a rate value. One of these two values is configurable as the status numeric output value. Rate type is configurable as fixed window, sliding window, or trigger type.
- **BooleanInputResistive** - BooleanPoint that reads the resistive input as one of two boolean states (equipment status). If the measured resistance level becomes higher than the threshold level, the point will output false.
- **BooleanInputVoltage** - BooleanPoint that reads the voltage input as one of two boolean states (equipment status). If the measured resistance level becomes higher than the threshold level, the point will output true.
- **BooleanInputCurrent** - BooleanPoint that reads the current input as one of two boolean states (equipment status). If the measured resistance level becomes higher than the threshold level, the point will output true.

Any non-Boolean selection results in a standard NumericPoint, but with a different type OnyxInputProxy extension. The BooleanInputPoint selections result in a BooleanPoint with an OnyxBooleanInputProxy extension. For more details, see “About Onyx Driver Proxy points” on page 14.

***Note:** After adding any Onyx Proxy point, you can edit name, port address, conversion, and facets if desired, but not type. To change type, you must delete the point and then add it again, selecting from the Type drop-down menu, as shown in **Figure 3-9**. The one exception here is the ResistiveInputPoint and ThermistorInputPoint, which are actually the same except for the conversion type used in the ProxyExt.*

Output type selection notes

JENEsys Edge 534 – N4 devices and Onyx extender modules have 10 digital outputs (DO) and 8 analog Output (AO) terminals. For any discovered output I/O terminal, the Add dialog preselects the appropriate writable OnyxPoint type, as either:

- **BooleanOutputWritable:** A standard BooleanWritable point, but with an OnyxBooleanOutput Proxy extension.
- **VoltageOutputWritable:** A standard NumericWritable point, but with OnyxVoltageOutput Proxy extension.

Unlike when adding universal input points, there is no alternate selection of type available when adding output points.

About Onyx Driver Proxy points

The onyxDriver's Proxy points are similar to other driver's Proxy points. You can add alarm and history extensions to them, and link them into other station logic as needed.

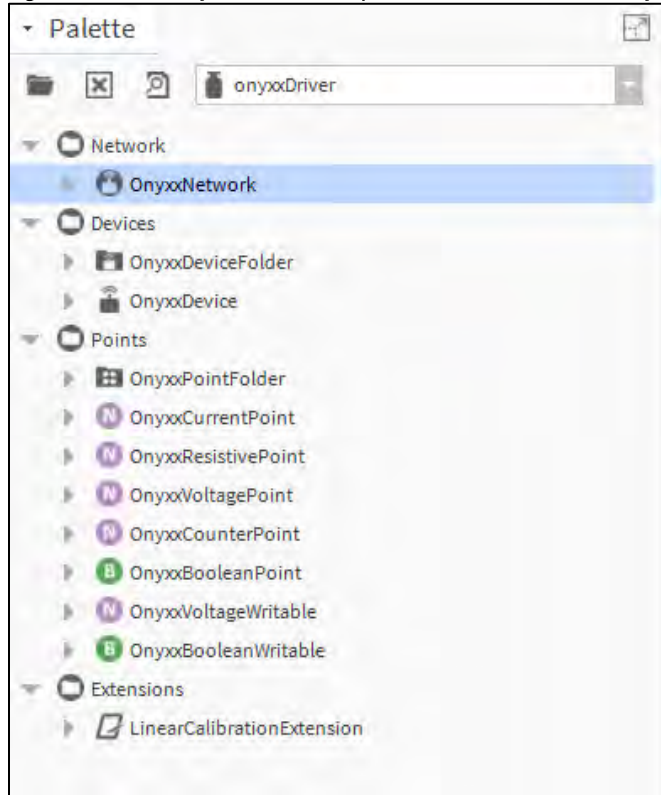
The following sections explain further:

- Onyx Driver point types
- Onyx Driver Proxy Ext common properties
- Conversion types in Onyx Driver Proxy points
- Scale and offset calculation (linear)
- Non-linear sensor support
- Linear Calibration Ext

Onyxx Driver point types

As seen in the onyxxDriver palette (Figure 3-10), there are 7 different onyxxDriver point types.

Figure 3-10 Onyxx Driver components as seen in the onyxxDriver palette



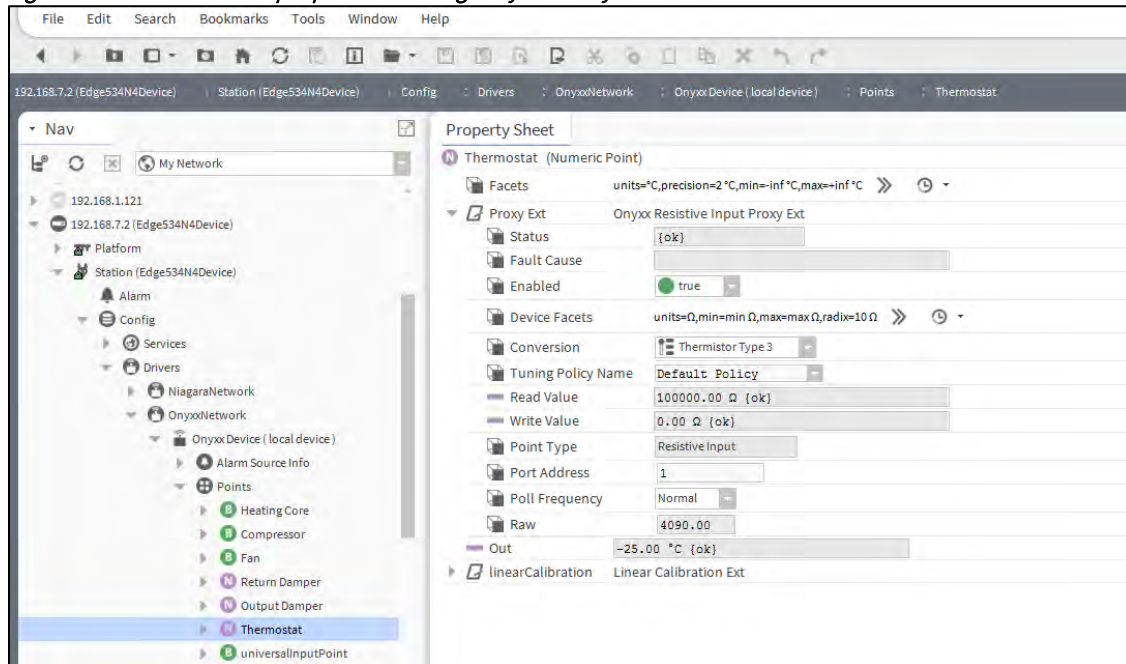
These 7 different **onyxxDriver** point types derive from the following 4 control point types:

- **BooleanPoint** - For Boolean “universal input” types; can be used to create BooleanInputPoint–Resistive point type
- **NumericPoint** - For Numeric “universal input” types; can be used to create CounterInputPoints, ResistiveInputPoints, ThermistorInputPoints (using ResistiveInputPoints and modifying), VoltageInputPoints, and CurrentInputPoints
- **BooleanWritable** - For Boolean output types; can be used to create BooleanOutputWritable Points
- **NumericWritable** - For numeric output types; can be used to create VoltageOutputWritable Points

Each **onyxxDriver** point type is typically unique by one or more properties in its Proxy extension. Also, the Proxy extension in each type contains the same (common) properties. Things unique to each **onyxxDriver** point types are explained in following sections, as well as common properties in each point’s Proxy extension.

Onyxx Driver Proxy Ext common properties

Figure 3-11 Common properties among Onyxx Proxy extensions



As shown in the property sheet view (Figure 3-11), each **onyxxDriver** point has the following properties in its Proxy extension (Proxy Ext):

- **Status** - (read only) This is the status of the Proxy extension.
- **Fault Cause** - (read only) If point falls into a fault status, this section will provide a text description as to why.
- **Enabled** – This can be either true (default) or false. When set to false, the point’s status becomes disabled and onyxxDriver polling is suspended.
- **Device Facets** - (read only) These are native facets used in Proxy read or writes (parent point’s facets are used in point status display, and are available for edit in Add and Edit dialogs in Point Manager).
- **Conversion** – This specifies the conversion used between the “read value” (in Device Facets) and the parent point’s output (in selected point facets). For details see “Conversion types in Onyxx Driver Proxy points.”
- **Tuning Policy Name** – This is the assigned tuning policy, which is typically “Default Policy.”
- **Read Value** - (read only) This is the last value read, using device facets.
- **Write Value** - (read only) This applies if writable point only. It is the last value written, using device facets.
- **Point Type** - (read only) This is the onyxxDriver point type, such as “BooleanResistiveInput,” “ResistiveInput,” “BooleanOutput,” and so on.
- **Port Address** - This is the point’s I/O terminal address for its hardware type (UI, AO, DO), automatically found during discovery. If port address is set to out of range or is a duplicate address (same address as same hardware type, same board), the point has a fault status.
- **Poll Frequency** - This is the assigned poll frequency group, either Slow, Normal (default), or Fast.
- **Raw** - (read only) This is the integer value returned from the **onyxxDriver** processor.

Conversion types in Onyxx Driver Proxy points

In any onyxxDriver Proxy point a Conversion property is available, with the following selections:

Note: Currently, Workbench provides no filtering of Conversion types based on the Onyxx Proxy point being configured. You will see the same Conversion drop-down selection list for a VoltageInputPoint as you do for a BooleanOutputWritable. However, only a few combinations are typically useful. In addition, there may be conversion types included that are not applicable, such as “Sm2 Flow Conversion,” which can be ignored.

500 Ohm Shunt

Applies only to a **VoltageInputPoint** used to read a 4-to-20 mA sensor, where the UI input requires a 500 ohm resistor wired across (shunting) the input terminals. The input signal is 2 V to 10 V. If this conversion type is not available, select Linear conversion instead.

As compared to a Linear or Generic Tabular conversion, this selection provides better resolution near the upper (20 mA/10 V) range of the input, compensating for input clamping protection the circuitry automatically applies when input voltage rises above 3.9 V.

Note: Currently, selection of 500 Ohm Shunt produces a “secondary” selection for conversion type, with all conversion types shown again. Typically, only the Linear type is valid, providing the 4-20 mA sensor has a linear response. In the Scale and Offset fields for this Linear conversion, you enter whatever calculated values are needed to display in the appropriate units of measurement for that sensor. **Figure 3-12** shows an example configuration of such a point.

Figure 3-12 Example VoltageInputPoint used for 4-to-20 mA input (Edit dialog from Point Manager)

Name	Type	Port Address	Poll Frequency	Conversion	Point Type	Facets
ui1_Lynx	VoltageInputPoint	3	Normal	500 Ohm Shunt	Voltage Input	units=V,precision=2,min=-inf,max=+inf

Name ui1_Lynx

Type Cannot edit

Port Address 3

Poll Frequency Normal

Conversion 500 Ohm Shunt Default

Point Type Voltage Input

Facets units=V,precision=2,V,min=-inf V,max=+inf V

OK Cancel

For background on this example configuration, see “Scale and offset calculation example 2”.

In the unusual case of a non-linear 4-20 mA sensor, provided you have the kitlo module installed, the secondary conversion type “Generic Tabular” can be used.

Default

(The default selection)

Conversion between “similar units” is automatically performed within the Proxy point. For example, for a **ThermistorInputPoint** if you set the units in its facets to degrees F, the point output automatically converts to the appropriate value, from degrees C.

If you set the parent point’s Facets to dissimilar units (often the case with a **VoltageInputPoint**), the parent point has a fault status to indicate a configuration error. In this case, you must select Linear conversion, and also set “Units” in the point’s “LinearCalibrationExt” to match the point’s facets. See “Scale and offset calculation (linear)” on page 18 and “Linear Calibration Ext” on page 20.

Generic Tabular

This Conversion type allows nonlinear support for devices other than for thermistor temperature sensors with units in temperature. Generic Tabular uses a “lookup table” method similar to the “Thermistor Tabular” conversion, but without predefined output units. Selection provides a popup **Tabular Conversion Dialog** in which you can enter a custom “source-to-result” non-linear curve, including the ability to import and export response curve files. For more details, see “Non-linear sensor support” on page 19 and “Curve File Import/Export notes” on page 26.

Linear

Applies to **VoltageInputPoint**, **ResistiveInputPoint**, and **VoltageOutputWritable** points. For these points, you typically want the point’s output value in some units other than Device Facets (voltage or resistance). The Linear selection provides two fields to make the transition:

- Scale: Determines linear response slope.
- Offset: Offset used in output calculation.

For related details, see “Scale and offset calculation (linear)” on page 18.

Note: For a **VoltageInputPoint** used for a 4-to-20 mA sensor (shunt resistor on UI input), another primary conversion type should be used, if available: 500 Ohm Shunt, with Linear typically selected as the “secondary” conversion. See more about that other conversion type above.

Reverse Polarity

Useful in a **BooleanInputPoint** or **BooleanOutputWritable** to reverse the logic of the hardware binary input or output.

Note: Be careful in the use of the reverse polarity conversion, as it may lead to later confusion when troubleshooting logic or communications problems.

Thermistor Type 3

Applies to a **ThermistorInputPoint**, where this selection provides a “built-in” input resistance-to-temperature value response curve for Type 3 Thermistor temperature sensors.

Tabular Thermistor

Applies to a **ThermistorInputPoint**, where this selection provides a control for a popup dialog for a custom resistance-to-temperature value response curve, including ability to import and export curve response files. For more details, see “Tabular Thermistor notes” on page 25 and “Curve File Import/Export notes” on page 26.

Scale and offset calculation (linear)

Unless you want a **VoltageInputPoint** or **VoltageOutputWritable** to operate with facets (units) of volts, you must set the point’s facets to the desired units, and set the ProxyExt Conversion property to Linear.

You must then enter your calculated scale and offset values in the Linear fields. Also, you must set the “Units” in its Linear Calibration Ext to match an input point’s facets—otherwise a fault status occurs.

Note: This also applies to any linearly-responding **ResistiveInputPoint** using point facets other than ohms.

Use the following formulas to calculate the Linear scale and offset values:

Where: x1 and x2 are the values (e.g. voltage), and y1 and y2 are the corresponding desired units.

- **Scale**
Scale = $(y2 - y1) / (x2 - x1)$
- **Offset**
Offset = $y1 - (\text{Scale} * x1)$

See example 1 and example 2.

Scale and offset calculation... Example 1

You have a 6-to-9 Vdc actuator to control with a **VoltageOutputWritable**, in terms of 0-to-100% input. Configure this point’s facets to have units: “misc,” “percent” (%).

Set the ProxyExt’s Conversion property to Linear, and enter these calculated scale and offset values:

- **Scale**
Scale = $(y2 - y1) / (x2 - x1)$
Scale = $(100\% - 0\%) / (9\text{ V} - 6\text{ V}) = 100/3 = 33.3333333$
- **Offset**
Offset = $y1 - (\text{Scale} * x1)$
Offset = $0\% - (33.3333333 * 6\text{ V}) = 0 - 200 = -200$

Scale and offset calculation... Example 2

You have a linear 4-to-20 mA differential pressure sensor to read with a **VoltageInputPoint** (using an external 500 ohm resistor wired across the input), reading effectively 2 V to 10 V. The range of this sensor is from -10 in. wc. to +10 in. wc. Configure this point’s facets to have units: “pressure,” “in/wc”.

Set the ProxyExt’s Conversion property to use “500 Ohm Shunt” (if available), then “Linear” as secondary, and in the Linear fields enter these calculated scale and offset values:

- **Scale**
 $Scale = (y2 - y1) / (x2 - x1)$
 $Scale = ((+10 \text{ in. wc.}) - (-10 \text{ in. wc.})) / (10 \text{ V} - 2 \text{ V}) = 20/8 = 2.5$
- **Offset**
 $Offset = y1 - (Scale * x1)$
 $Offset = -10 \text{ in. wc.} - (2.5 * 2 \text{ V}) = -10 - 5 = -15$

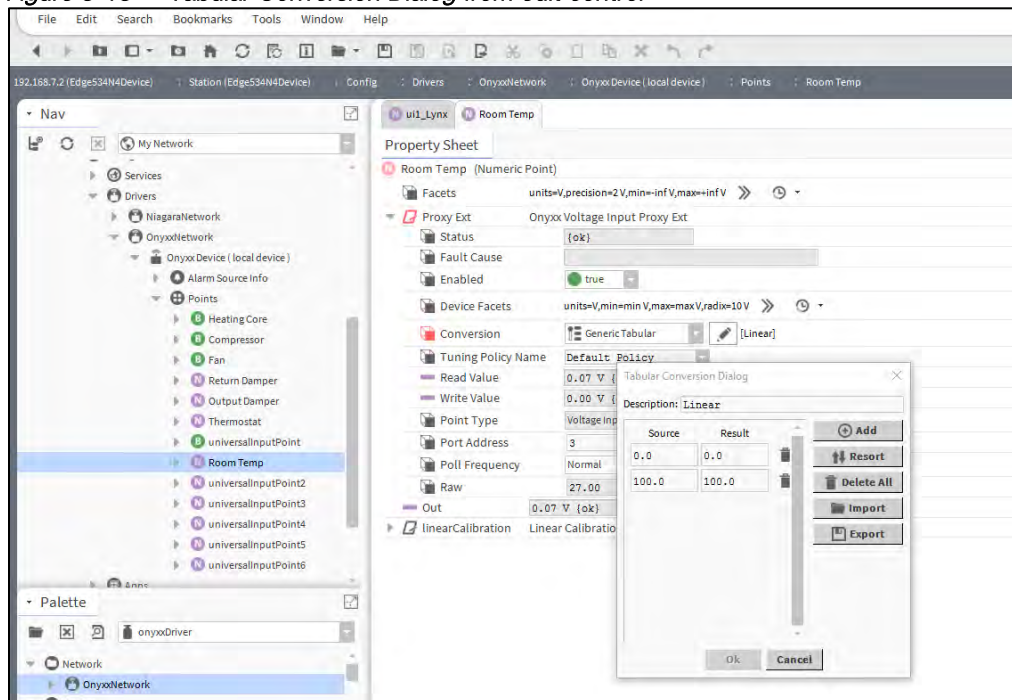
Figure 3-12 shows this point’s configuration in a Workbench dialog. Note that the Linear Calibration Ext for this point will require its “Units” property to be configured to match the facets for this point. Otherwise, the point will have a fault status. See the “Linear Calibration Ext” section for related details.

Non-linear sensor support

Provided that the kitlo module is installed, the conversion selection Generic Tabular provides a way for a **VoltageInputPoint** or **ResistiveInputPoint** to support a nonlinear signal on a universal input (using point Facets other than temperature). Note that the **ThermistorInputPoint** also provides non-linear support for resistance-based temperature sensors, but point Facets are limited to temperature.

If you select Conversion type of “Generic Tabular” an edit control (note pad icon) appears in the property sheet beside it. Click the icon to see the Tabular Conversion Dialog, as shown in **Figure 3-13**.

Figure 3-13 Tabular Conversion Dialog from edit control



This dialog allows you to edit the current “source-to-results” non-linear curve used by the Proxy point, import another curve (.xml file), or export (save) the current curve as an .xml file. In this way you can provide any custom non-linear curve needed. **Figure 3-13** shows a point with a non-linear curve described in the “Non-linear sensor example” below.

The following notes apply to using this feature:

- A curve requires at least 2 points (rows), each using “Source” to “Results” values.
 - ❖ Source values use the “Device Facets” of the Onyx Proxy point, and must be in the range of the controller input. Therefore, if a **VoltageInputPoint**, source values must be between 0 V to 10 V, or if a **ResistiveInputPoint**, source values must be between 0 to 100,000 (Ohms).

- ❖ Result values are typically the measured value, regardless of units, at each source point. This assumes no further scaling using a LinearCalibrationExt—i.e., its Scale property is the default “1”.
- Points are used in ascending order, by the Source column.
- You can add as many points as is needed (there is no hard-coded limit). Click the **Add** button to add a new row for a point.
- If you skip a point, just add it at the end, and then click **Resort**. This automatically reorders all points in ascending order, by the Source column.
- Click the **delete** icon (trash can) beside any point to delete it from the curve. If you click the **Delete All** button, all points (plus any Description text) is removed (typically, you do this only to enter a new non-linear curve).
- Description text appears in the ProxyExt property sheet, beside the Conversion edit icon.
- When you click **Save**, the curve configuration is stored as part of the conversion object in the station (there is no association with any external file, in case you imported a non-linear curve).

Note: If you update a non-linear curve (say, add a point) after it was imported in one or more OnyxPoints, you need to export it (save) and re-import it in other points, in order for them to be updated.

Non-linear sensor example

You have a photo resistor-based illuminance sensor that measures light output from 0 to 800 lux, supplying a resistance of 30 K ohms to 3 K ohms. Using a **ResistiveInputPoint**, select the conversion type Generic Tabular, and click the edit control for the Tabular Conversion Dialog (see **Figure 3-13** on page 19).

In the dialog, enter the sensor’s non-linear response curve (shown below). Configure this point’s facets to have units: “illuminance,” “lux” (lx), and the same units in the point’s Linear Calibration Ext.

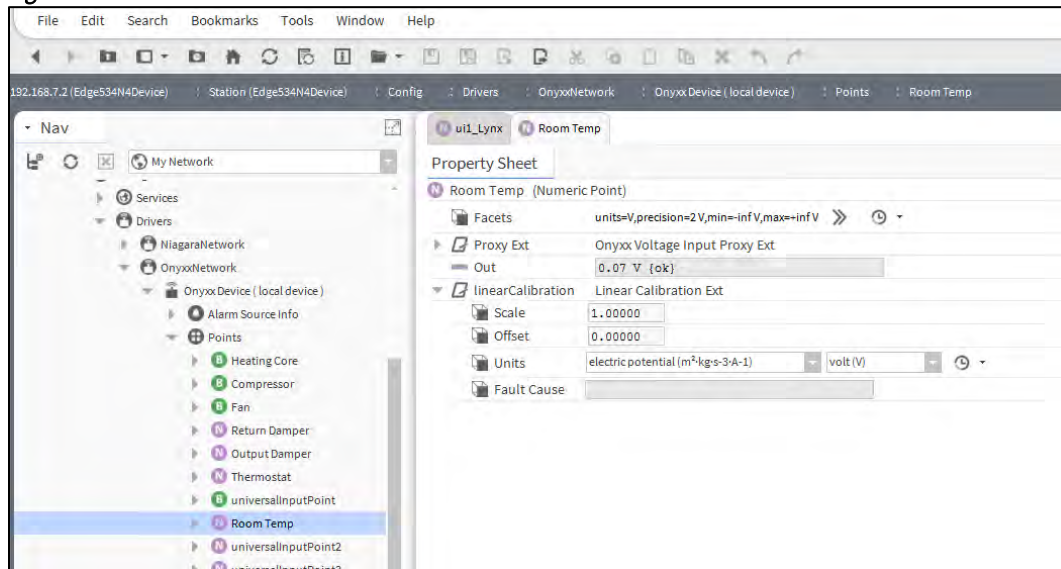
Table 3-1 Example illuminance sensor source (ohms) to results (lux)

Ohms	Lux	Ohms	Lux
3000	800	10200	100
3200	700	10900	90
3500	600	11600	80
4000	500	12400	70
4200	450	13300	60
4600	400	14400	50
5000	350	15800	40
5500	300	17700	30
6100	250	20300	20
6900	200	30000	0
8200	150		

Linear Calibration Ext

By default, three of the UI-type OnyxPoints (**ResistiveInputPoint**, **ThermistorInputPoint**, **VoltageInputPoint**) include a “linearCalibration” slot containing 4 properties, as shown in **Figure 3-14**.

Figure 3-14 Linear Calibration Extension



These properties allow you to “calibrate” the calculated value before it is applied to the Out slot, where: [(calculatedValue x Scale) + Offset] = Out value.

Usage is optional, although Offset and Units are commonly configured.

Note: In most cases where the parent Onyx Proxy point's facets have been edited from defaults, note you must edit the Units value in the Linear Calibration Ext to match the units in the point facets, otherwise the parent Proxy point will have a fault status!

Typically, you see this fault status immediately after you add a new input point, for example a **VoltageInputPoint** or **ResistiveInputPoint**, and configure it with a Linear conversion type (including a scale and offset), and then specify the point's facets. It may not be immediately clear that the problem is in this Linear Calibration Ext, where you must match its Units value to the units in the point's facets.

- **Scale** - Default is 1.0. Typically, you leave this at default. One exception is if you copied the LinearCalibrationExt under a **CounterInputPoint**, in order to get a scaled total.
- **Offset** - Default is 0.0. Can be either a positive or negative value, as needed. Often, this is useful to compensate for signal errors introduced by sensor wiring resistance.
- **Units** - Set this to the same units as in the parent Proxy point's facets (see Note: above).
- **Fault Cause** – This provides a reason whenever this extension causes the parent Proxy point to be in fault, e.g.: Units between point and extension are not convertible.

Note: You can also copy the linearCalibration extension without error to other OnyxPoints based on Numeric points. For example, a **CounterInputPoint** where a “scaled total” out value is desired. Consider a **CounterInputPoint** for a flow rate meter that provides a contact closure for every 0.15 gallons, configured as:

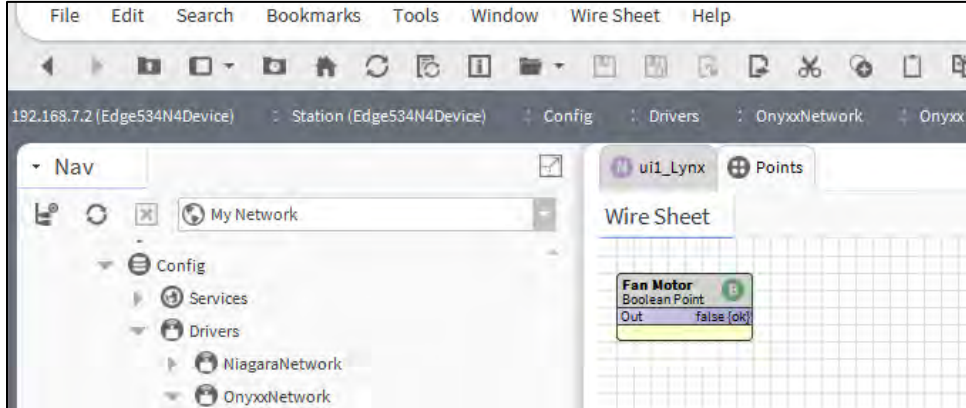
- Facets: gallons [units: volume (m³), gallon (gal)]
- OnyxCounterInputProxyExt:
 - ❖ Conversion: Default
 - ❖ Output Select: Count
 - ❖ Rate Calc Type: FixedWindowRateType
 - ❖ Rate Calc:
 - Scale: 9.0
 - Interval: 15 s
- LinearCalibrationExt:
 - Scale: 0.15
 - Offset: 0.0

By copying a *LinearCalibrationExt* to the *CounterInputPoint* and specifying the item quantity/pulse as the *Scale* value (in this case, 0.15), the count output value will read in gallons, rather than number of pulses. Note that the “rate calc” setup provides the “gallons per minute” scaling that will reflect in the Rate slot of the point’s *ProxyExt*, where *Scale* was calculated as $60 \text{ (sec/min)} * 0.15 \text{ gal/pulse} = 9$.

BooleanInputPoint

A *BooleanInputPoint* is a *BooleanPoint* with *OnyxxBooleanInputProxyExt*. It configures a UI to read contact closures as a status boolean (equipment status, binary input, BI).

Figure 3-15 Onyxx Boolean Input Point



The *BooleanInputPoint* contains an additional property named “Boolean Threshold”. This property is hidden by default. This property allows the user to specify when the boolean point should alter its value (true/false). To unhide the Threshold so as to modify its value, do the following steps:

- Step 1: In Nav window, double click the points Proxy Ext.
- Step 2: In main window, change the view to be “Slot Sheet.”
- Step 3: Right-click the line (from table) that holds the “booleanThreshold” property and select **config flags**.
- Step 4: De-select “Hidden” and click **OK**.
- Step 5: Change the view back to property sheet to see/modify.

BooleanInputResistive

Inputs read a 0-100,000 ohm resistive signal. If the signal is above the threshold (50% of Raw Value by default), the Boolean value of false is reported. If the signal is below the threshold, the Boolean value of true is reported. The threshold value can be changed by displaying and editing it.

BooleanInputVoltage

Inputs read the voltage signal of self-powered 0-10 Vdc devices. If the signal is above the threshold (50% of Raw Value by default), the Boolean value of true is reported. If the signal is below the threshold, the Boolean value of false is reported. The threshold value can be changed by displaying and editing it.

BooleanInputCurrent

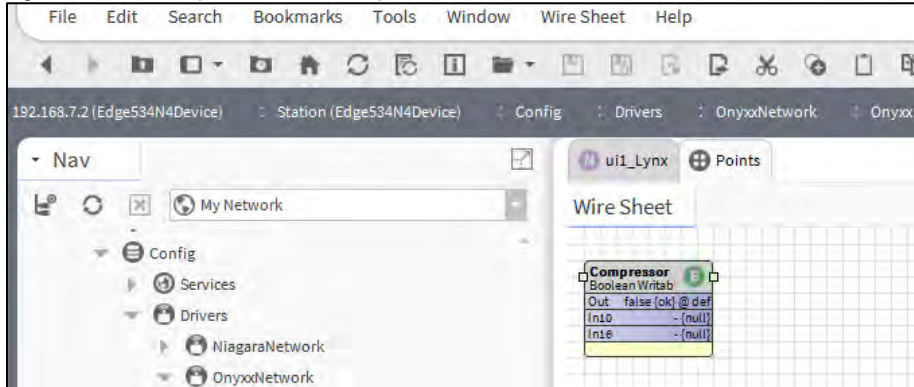
Inputs read the current signal of self-powered 4-20mA devices. If the signal is above the threshold (50% of Raw Value by default), the Boolean value of true is reported. If the signal is below the threshold, the Boolean value of false is reported. The threshold value can be changed by displaying and editing it.

Note: The default operation is normal logic (closed contacts at UI is output value “true”). If needed, you can “flip” this logic in the *ProxyExt* by assigning a Conversion type of “Reverse Polarity.” This causes the “reversed” Boolean state at the output value. Typically, you monitor normally open (N.O.) equipment contacts using normal logic.

BooleanOutputWritable

A BooleanOutputWritable is a BooleanWritable with OnyxxBooleanOutputProxyExt. It represents a digital output (DO). Depending on I/O platform, this may be a relay output or triac output. All standard BooleanWritable features apply, including right-click actions, priority input scheme, and minimum on/off properties.

Figure 3-16 Onyxx Boolean Output Writable



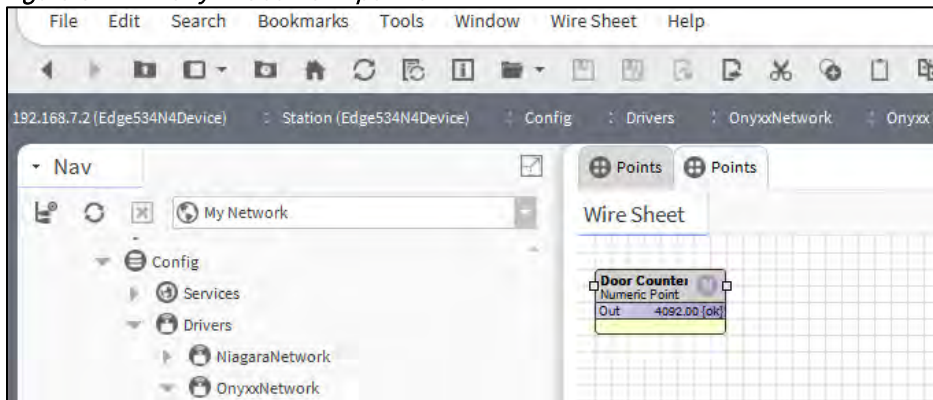
Note: The default operation is normal logic (closed contacts at DO if input value “true”). If needed, you can “flip” this logic in the ProxyExt by assigning a Conversion type of “Reverse Polarity.” This causes the “reversed” boolean state going from input to output.

CounterInputPoint

A CounterInputPoint is a NumericPoint with OnyxxCounterInputProxyExt. The Counter Input Point is used in the station database to count dry-contact pulses with a change-of-state (COS) frequency of up to 50 Hz and a 50% duty cycle. These points will increment on both sides of a pulse, therefore a full cycle will be 2 counts. If a single change in value is desired, the user will need to divide the result by 2.

Note: Minimum dwell time must be > 2 ms.

Figure 3-17 Onyxx Counter Input Point



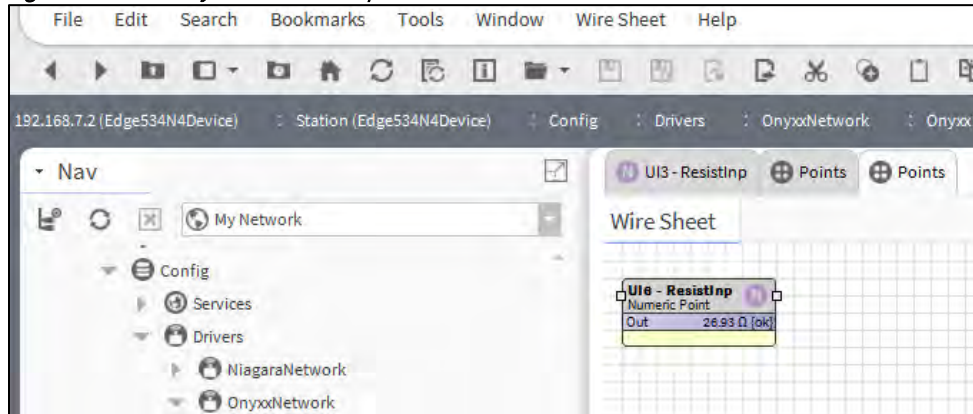
ResistiveInputPoint

A ResistiveInputPoint is a NumericPoint with OnyxxResistiveInputProxyExt. It configures a UI to read a resistance from 0 to 100,000 ohms. Configuration provides for default (ohms).

- If a linear-responding device and other units are needed, select Conversion type Linear—see “Scale and offset calculation (linear)” on page 18.
- If a non-linear device and other units are needed, select Conversion type Generic Tabular—see “Non-linear sensor support” on page 19.

Note: If using a thermistor temperature sensor, see *ThermistorInputPoint*.

Figure 3-18 Onyx Resistive Input Point



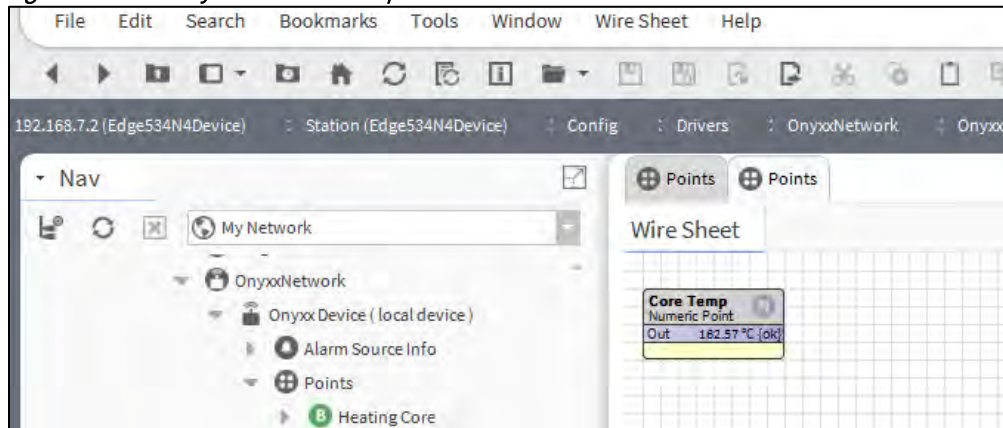
The ResistiveInputPoint's Proxy Ext contains only common Onyx Proxy Ext properties. Among these, note that the read-only "Device Facets" property is preset to ohms.

ThermistorInputPoint

A ThermistorInputPoint is a NumericPoint with OnyxResistiveInputProxyExt configured for a UI to read a thermistor temperature sensor. It is pre-configured with a linearCalibration extension, for offset correction. For details, see "Linear Calibration Ext" on page 20.

Note: A ThermistorInputPoint is the same as a ResistiveInputPoint, only with a different "Conversion" setting in its ProxyExt. However, the Onyx Point Manager lists this point separately, so it is covered here separately.

Figure 3-19 Onyx Thermistor Input Point



The point's Proxy Ext contains common OnyxProxyExt properties. Among those properties, note that the read-only "Device Facets" is preset to ohms.

Depending on the type of thermistor temperature sensor you are using, you can select a Conversion type of either Type 3 Thermistor, or any other using a Tabular Thermistor conversion.

Type-3 Thermistor

Table 3-2 shows the hard-coded response curve used if ProxyExt property Conversion is set to "Thermistor Type 3":

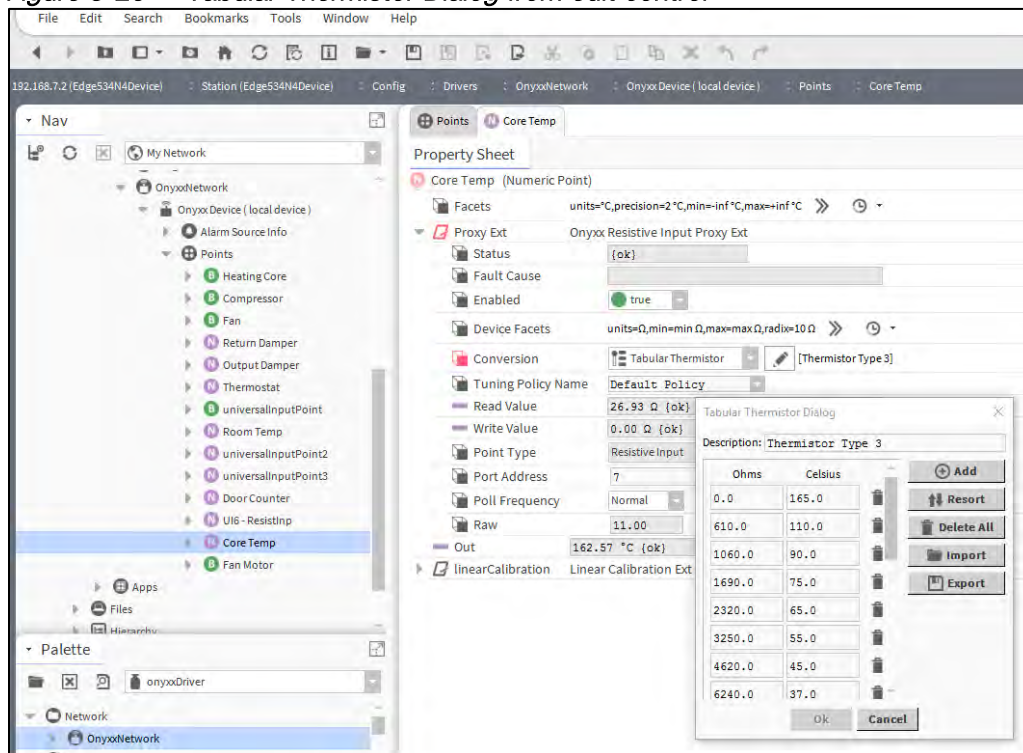
Table 3-2 Type-3 Thermistor Curve (from Ndio)

Ohms	C °	Ohms	C °
0	165	12268	20
610	110	15136	15
1060	90	18787	10
1690	75	23462	5
2320	65	29462	0
3250	55	37462	-5
4620	45	47549	-10
6240	37	61030	-15
8197	30	78930	-20
10000	25	100000	-25

Tabular Thermistor notes

In the ProxyExt of the thermistor point, if you select Conversion type of “Tabular Thermistor,” an edit control (note pad icon) appears in the property sheet beside it. Click the icon to see the Tabular Thermistor dialog, as shown in Figure 3-20.

Figure 3-20 Tabular Thermistor Dialog from edit control



This dialog allows you to edit the current ohms-to-degrees Celsius curve used by the Proxy point, import another thermistor curve (.xml file), or export (save) the current thermistor curve as an .xml file. In this way you can provide any custom thermistor curve needed.

The following notes apply to using this feature:

- A curve requires at least 2 points (rows), each using ohms to degrees Celsius values.
- Points are used in ascending order, by the “ohms” column.
- You can add as many points as is needed (there is no hard-coded limit). Click the **Add** button to add a new row for a point.
- If you skip a point, just add it at the end, and then click **Re-sort**. This automatically reorders all points in ascending order, by the “ohms” column.
- Click the **delete** icon (trash can) beside any point to delete it from the curve. If you click the **Delete All** button, all points (plus any Description text) is removed (typically, you do this only to enter a new thermistor curve).
- Description text appears in the ProxyExt property sheet, beside the Conversion edit icon.
- When you click **Save**, the curve configuration is stored as part of the conversion object in the station (there is no association with any external file, in case you imported a thermistor curve).

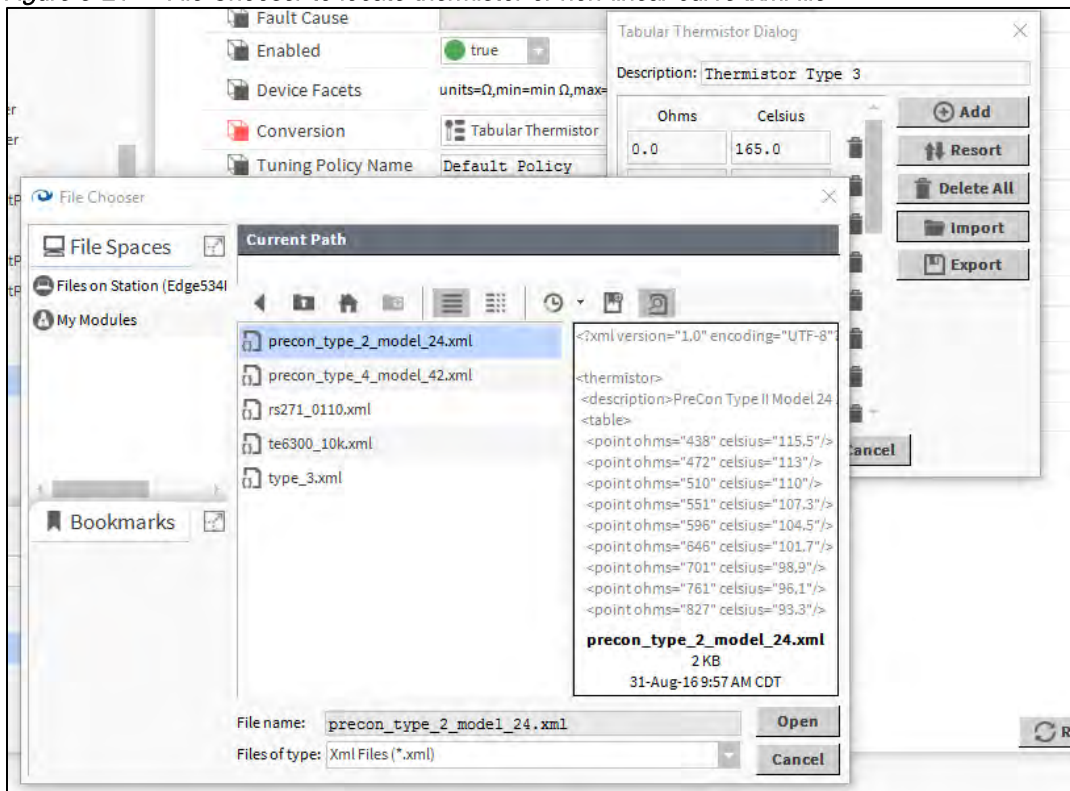
Note: If you update a thermistor curve (say, add a point) after importing it into one or more Onyx points, you need to import it in those points again, in order for them to be updated.

Curve File Import/Export notes

As needed, import (load) or export (save) a thermistor or non-linear curve file in xml format using the Import and Export buttons in either the Tabular Thermistor Dialog (Figure 3-20) for a “Tabular Thermistor” conversion, or Tabular Conversion Dialog (Figure 3-13 on page 19) for a “Generic Tabular” conversion.

When you do this, the standard Niagara 4 File Chooser dialog appears, as shown in Figure 3-21.

Figure 3-21 File Chooser to locate thermistor or non-linear curve .xml file



To import a thermistor curve file, you can navigate under your “My Modules” folder to the **xml** folder in the **kitlo** module (Jar) file (if available, see Figure 3-21). If the **kitlo** module is not available, navigate to the **xml** folder in the **Ndio** module (Jar) file.

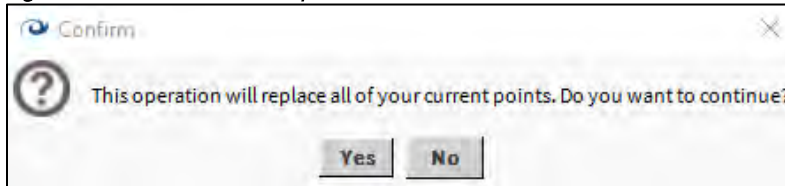
Or, you can navigate to any other location you have a thermistor curve file. Import any other saved, custom non-linear conversion .xml files the same way—see “Non-linear sensor support” on page 19.

As of the date of this document, the **kitlo** module contains an **xml** folder with five (5) different thermistor curves in xml format, as follows:

- precon_type_2_model_24.xml — For Precon type 2 model 24 sensor
- precon_type_4_model_42.xml — For Precon type 4 model 42 sensor
- rs271_0110.xml — For Radio Shack sensor model 271-0110
- te6300_10k — For TE-6300 10 K type sensor
- type_3.xml — For standard Type-3 Thermistor sensor

Note: Upon any curve file import, all existing thermistor or non-linear curve configuration for that Onyx Proxy point is overwritten (replaced by configuration in the imported file). A confirmation dialog advises you of this before you import, as shown in **Figure 3-22**.

Figure 3-22 Curve file import confirmation



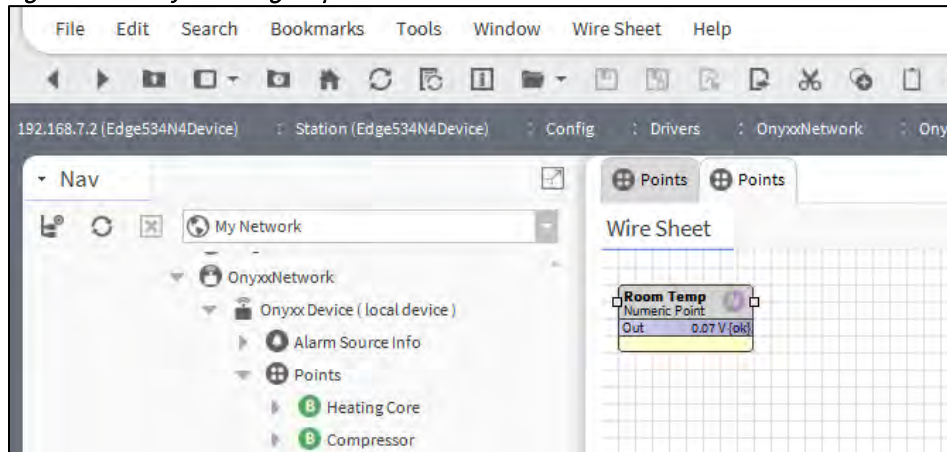
To export a curve .xml file, use the File Chooser to navigate to any location you wish to save the current thermistor curve configuration, and supply a File Name. You can use this saved .xml file in the configuration of any other Onyx or Ndio Proxy point, whether in this station or another station.

VoltageInputPoint

A VoltageInputPoint is a NumericPoint with OnyxVoltageInputProxyExt. It configures a UI to read a Vdc signal from 0- to-10 V. Currently, configuration provides for reading in volts or a linear response in other units, by selecting “Linear” as the conversion type. See “Scale and offset calculation (linear)” on page 18.

The VoltageInputPoint is pre-configured with a linearCalibration extension, to allow for offset correction. For details, see “Linear Calibration Ext” on page 20.

Figure 3-23 Onyx Voltage Input Point

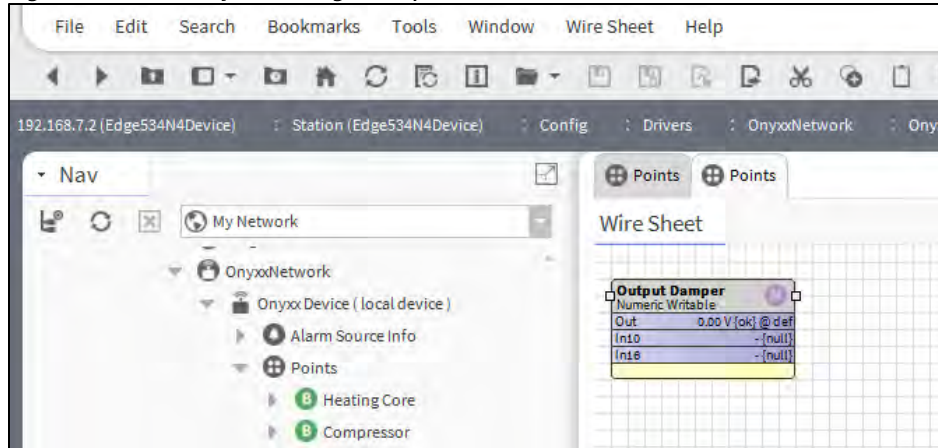


The VoltageInputPoint’s Proxy Ext contains only common OnyxProxyExt properties. Among these, note the read-only “Device Facets” property is preset to volts (V).

VoltageOutputWritable

A VoltageOutputWritable is a NumericWritable with OnyxxVoltageOutputProxyExt. It represents a 0-to-10 Vdc analog output (AO). All standard NumericWritable features apply, including right-click actions and priority input scheme.

Figure 3-24 Onyxx Voltage Output Writable



The point's Proxy Ext contains common OnyxxProxyExt properties. Among those properties, note that the read-only "Device Facets" is set to units Volts (V). For operation in units other than volts, see "Scale and offset calculation (linear)" on page 18.

Onyxx Network Driver Plugin Guides

Plugins provide views of components, and can be accessed many ways. For example, double-click a component in the tree for its default view. In addition, you can right-click a component, and select from its "views" menu. For summary documentation on any view, select Help > On View (F1) from the Workbench menu, or press F1 while the view is open.

Onyxx Network Driver Plugin Guides Summary

Summary information is provided on views specific to components in the onyxxDriver module, as follows:

- N Device Manager
- Onyxx Driver Point Manager

N Device Manager

Use the N Device Manager to add, edit, and access Onyxx Devices that represent physical modules of I/O on the host JENEsys Edge 534 – N4 controller platform. The N Device Manager is a view on the onyxxDriver Network. To view, right-click onyxxNetwork and select Views > N Device Manager.

Tables in the Onyxx Device Manager view have the following columns:

- Name - Component name of the I/O board
- Type - Niagara 4 type (currently there is only Onyxx Device)
- Exts - Shortcut to the device extensions (only one extension – Points)
- Status - Current status of the I/O board
- Device Path - Defines the virtual File path that represents a device

For more details, see "Onyxx Device Manager" on page 9.

Onyxx Driver Point Manager

Use the onyxxDriverPointManager to add, edit, and access Onyxx Proxy points under the onyxxPointDeviceExt extension of a selected Onyxx Device, or in an onyxxPointFolder. The onyxxDriverPointManager is the default view on both of those components. To view, right-click the onyxxPointDeviceExt extension or onyxxPointFolder and select **Views > Onyxx Driver Point Manager**.

Tables in the onyxxDriverPointManager view have the following columns:

- **Name** - Component name of the Onyxx Proxy point
- **Type** - Onyxx point type of the configured point
- **Port Address** - Hardware I/O address of the point
- **Poll Frequency** - Speed in time point will be polled
- **Conversion** - The Conversion that is applied to the point
- **Point Type** - Onyxx Proxy Extension Type of the configured point
- **Address** - Hardware I/O address of the point
- **Out** - Current value of the point
- **Facets** - defined point facets

For more details, see “Onyxx Point Manager” on page 12.

JENEsys Edge Troubleshooting

Below information is meant to provide some basic troubleshooting before needing to call into support.

Platform Connection Issues

- Make sure the JENEsys Edge 534 – N4 is fully up and running. Its power LED should turn on and its heartbeat LED should be flashing.
- Make sure the Ethernet cable is connected firmly to the Ethernet port on the JENEsys Edge 534 – N4. The LEDs on the Ethernet port will indicate if the JENEsys Edge 534 – N4 is connected to the network.
- You can also view the Ethernet Activity LEDs (Tx, Rx) which will indicate the JENEsys Edge 534 – N4 is transmitting and receiving on the Ethernet network.
- If you are connecting directly from your computer to the JENEsys Edge 534 – N4, ensure your computer's network settings are configured with a unique IP address within the same subnet as the device.
- If you are connecting to the JENEsys Edge 534 – N4 on a network make sure your computer's network settings are not set for DHCP.
- Close and re-open the browser to make sure that the browser did not cache the previous page.

Onyx Network Issues

❖ *Slow or no communication*

Adding a 120 ohm resistor to the onyxNetwork connector may fix this. Adding a resistor will not harm anything. It will either fix the communication or is related to something else.

Primary Ethernet Port Doesn't work

Ensure all wiring is connected before powering the device. This includes the Ethernet cable. A power cycle may correct this.

Communication is slow for BACnet

Ensure you have a 180 ohm resistor on the RS-485 connector.

Copyright ©2017. Lynxspring, Inc. All Rights Reserved.

NOTICE: All information contained herein is, and remains the property of Lynxspring, Incorporated.

The intellectual and technical concepts contained herein are proprietary to Lynxspring, Incorporated and may be covered by U.S. and Foreign Patents, patents in process, and are protected by trade secret or copyright law. Dissemination of this information or reproduction of this material is strictly forbidden unless prior written permission is obtained from Lynxspring Incorporated.

Lynxspring[®], JENEsys[®], Onyx[®] and Helixx[®] are registered trademarks of Lynxspring, Inc.

JENEsys Edge[™] is a trademark of Lynxspring, Inc.

Niagara Framework[®] is a registered trademark of Tridium, Inc.

The information and/or specifications published here are current as of the date of publication of this document. Lynxspring, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters in Lee's Summit, Missouri. Products or features contained herein are covered by one or more United States or foreign patents. Other brand and product names are trademarks or registered trademarks of their respective holders. This document may be copied by parties who are authorized to distribute Lynxspring 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 Lynxspring, Inc. Complete Confidentiality, Trademark, Copyright and Patent notifications can be found at:

<http://resources.lynxspring.com>.