Technical Document

CCN Driver Guide

October 7, 2021

niagara4

CCN Driver Guide

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

Confidentiality

Trademark notice

Copyright and patent notice

Contents

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. To make the most of the information in this book, readers should have some training or previous experience with Niagara software, as well as experience working with JACE network controllers.

Document content

This document provides information about the CCN driver, components and plugins, license tools and other topics related to the CCN 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

Changes to this document are listed in this topic.

October 7, 2021

Removed chapter "ComfortVIEW tunneling through a controller" including sections "Configuring the controller" and "Configuring the ComfortVIEW station". Deleted component "ccn-CcnTunnelHelper" from the "Components" chapter.

September 3, 2021

Initial release of this document.

Related documentation

Every standalone driver guide relies on this guide for the explanation of common driver properties, concepts, and tasks.

- *Getting Started with Niagara (User Guide)*
- *Niagara Platform Guide*
- *Niagara Station Security Guide*
- *Niagara Alarms Guide*
- *Niagara Scheduling Guide*
- *Niagara Provisioning Guide*
- *Niagara System Database and System Indexing Guide*
- *Niagara Histories Guide*
- *Niagara LDAP Guide*

Chapter 1 Overview

Topics covered in this chapter

♦ Quick start
♦ CCN Architecture
♦ License
♦ Limitations and changes

The Carrier Communication/Comfort Network (CCN driver) integrates CCN devices and data into the Niagara Framework environment.

The CCN driver is a serial driver with three primary components:

- The **CcnNetwork**
- The **CcnDevice**
- A collection of objects (proxy points) that shadow I/O and variables in the network.

Quick start

Follow these steps to get started with CCN driver.

Prerequisites:

You are working in Workbench running on a PC and are connected to a remote station.

Step 1 Open the **ccn** palette, copy the **CcnNetwork** component to the **Drivers** node, right-click the **CcnNetwork** node and click **Views→Ccn Network View**.

The **Ccn Network View** (a property sheet) opens.

Step 2 Update at least these properties and click **Save**.

- Bus and element ranges: **First Bus No**, **Last Bus No**, **Low Elem No** and **High Elem No** for the devices to discover.
- The driver's **Bus Address** and **Elem Address** (typically 230-238).
- **Comm Port**.

NOTE: On a controller, configure only the available ports.

Step 3 Add devices by opening the **Ccn Device Manager** view, click **Discover**, then select and add the CCN device(s).

Step 4 Once device(s) are added, double-click the **Points** folder, which is available under **CcnDevice**.

The **Ccn Table Manager** view opens.

Step 5 To discover the table groups available under a device, click **Discover**, then select and add the groups to the database.

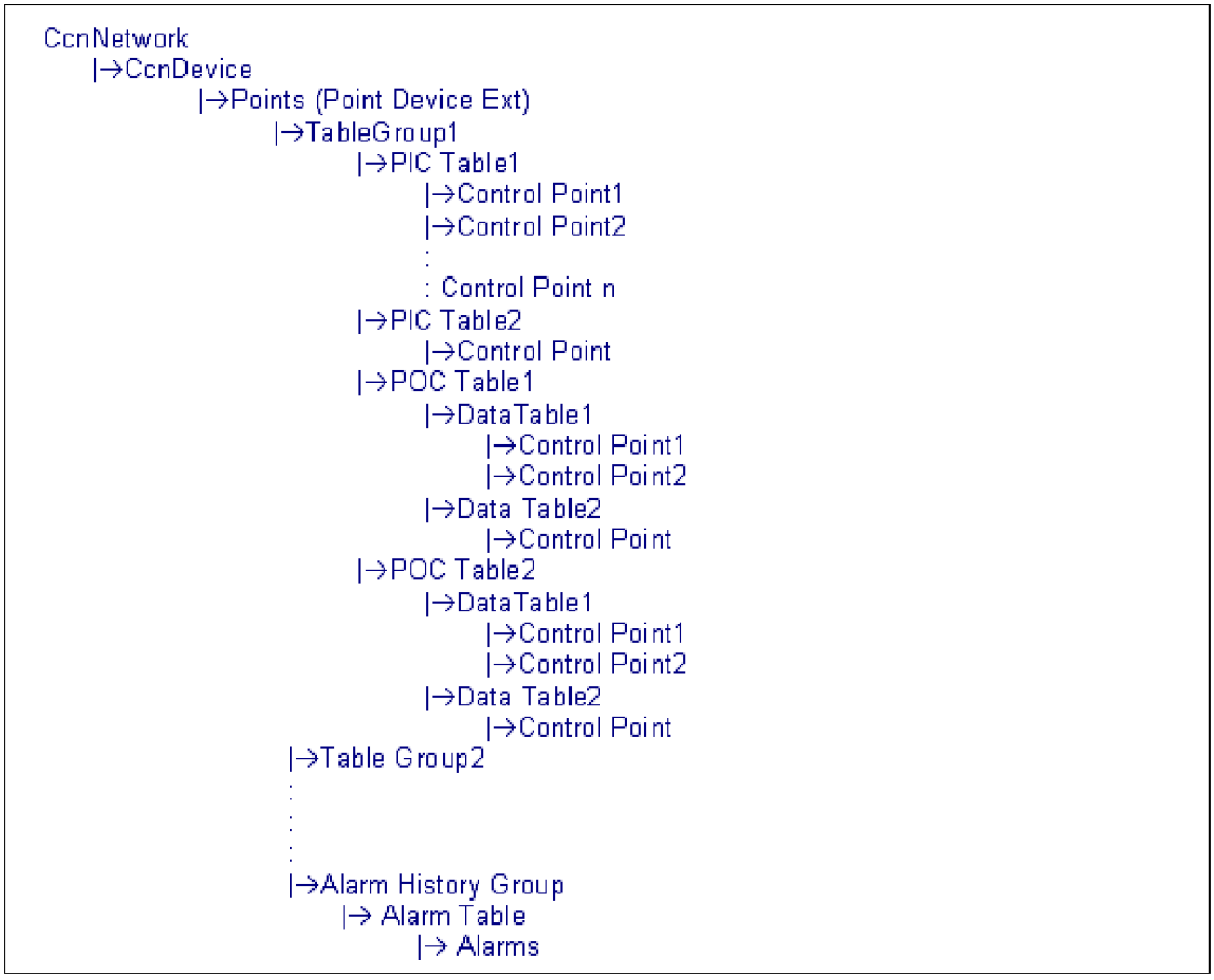
The driver adds the group tables and categorizes them.

Step 6 Once the table group(s) are added, navigate to a table (PIC, POC or FID) under a table group and discover points.

CCN Architecture

The CCN driver uses the standard Niagara network architecture with **CcnDevices** under a **CcnNetwork**. Normally, drivers have direct points under a **Points** extension. The CCN driver has table groups under a **Points** extension with points under the table groups. Diagrammatically the hierarchy looks like this:

Figure 1 Ccn driver architecture



The CCN driver provides support for several types of data:

- **CcnTableGroup** is a container within which to organize CcnTable proxy points.
- **CcnPicTable** contains CCN PIC- type proxy points.
- **CcnPocTable** contains CCN POC-type proxy points.
- **CcnDataTable** contains CCN Data-type proxy points.
- **CcnDataTablewithTimeSchedule** contains special proxy points for the **CCN DataTable** time schedule that can be represented either in tabular form, as other tables are represented, or graphically as a time schedule.
- **CcnFidTable** contains CCN Fid-type proxy points.

- **CcnFidTablewithTimeSchedule** contains special proxy points for the **CCN FidTable** time schedule that can be represented either in tabular form, as other tables are represented, or graphically as a time schedule.
- **CcnAHTable** contains CCN Alarm History-type proxy points.
- **CcnInputProxy** shadows the behavior of the CCN Input Points under a **CcnPicTable**.
- **CcnOutputProxy** shadows the behavior of the CCN Output Points under a **CcnPicTable**, **CcnDataTable** with **CcnTimeSchedule**, **CcnFidTable** or **CcnFidTable** with a **CcnTimeSchedule**.

License

The CCN driver supports one of two operating modes. The default mode must have feature **ccnl** in the license file. For the extended mode, the license feature must include both **ccn** and **ccnl** in the license file.

The extended mode driver is not offered for sale at this time.

"ccnl"——— ccn standard license

"ccn + ccnl" ——— ccn extended license

A CCN Standard (license feature **ccnl**) provides:

- Read/write/force/auto display of table entries
- Read/write set point table entries
- Read/write time schedule table entries
- Discovery (learn/create) support for display, set point and time schedule tables
- Upload/download support for display, set point and time schedule tables
- Alarm handling (display and logging)
- Broadcast date/time, broadcast acknowledger, alarm broadcast acknowledgment support
- Device status support

A CCN Extended (license feature **ccn** + **ccnl**) provides all the above plus:

- Additional support for Read/write/force/auto maintenance of table entries
- Additional support for Read/write configuration tables
- Additional support for Discovery (learn and create) for maintenance and configuration tables
- Upload and download support for maintenance and configuration tables

Limitations and changes

Time-related CCN points require a value in the format: hh:mm. The driver displays no error messages in the user interface. It does show an error message in the console. The driver supports a single **CcnNetwork** per station and was tested with a single network trunk per station.

Testing has not been performed for these functions:

- Alarm acknowledgement
- Time broadcasting and broadcast acknowledgement
- Upload and download options from the ComfortVIEW tool in a remote controller
- Two comm controller ports

Chapter 2 Network and device management

Topics covered in this chapter

- ♦ Adding a CcnNetwork from the palette
- ♦ Adding a CcnNetwork using the New button
- ♦ Discovering and adding CcnDevices, tables and points
- ♦ Adding a CcnDevice from the palette
- ♦ Adding a CcnDevice using the New button
- ♦ Designating a station as a CCN alarm or broadcast acknowledger
- ♦ Designating a station as a CCN time broadcaster

Configuring the network involves discovering tables and points.

Adding a CcnNetwork from the palette

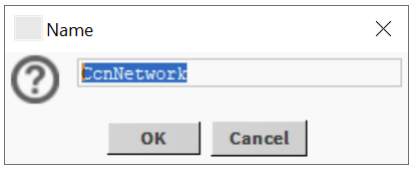
The **CcnNetwork** component is the parent folder for **CcnDevices**, tables and points.

Prerequisites: You are using Workbench in a PC and are connected to a remote station.

Step 1 Open the **ccn** palette.

Step 2 Copy the **CcnNetwork** component to the **Drivers** node in the Nav tree.

The **Name** window opens.



Step 3 Give the network a name or use the default name and click **OK**.

Step 4 Right-click the **CcnNetwork** node and click **Views→Ccn Network View**.

The **Ccn Network View** opens.

Step 5 Update at least these properties and click **Save**.

- Bus and element ranges for the: **First Bus No**, **Last Bus No**, **Low Elem No** and **High Elem No** for the devices to discover.
- The driver's **Bus Address** and **Elem Address** (typically 230-238).
- The **Comm Port**.

NOTE: On a controller, configure only the available ports.

As with most other drivers, the status of a **CcnNetwork** is either `{Ok}` or less typical `{fault}` (`{fault}` may result from a licensing error). 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.

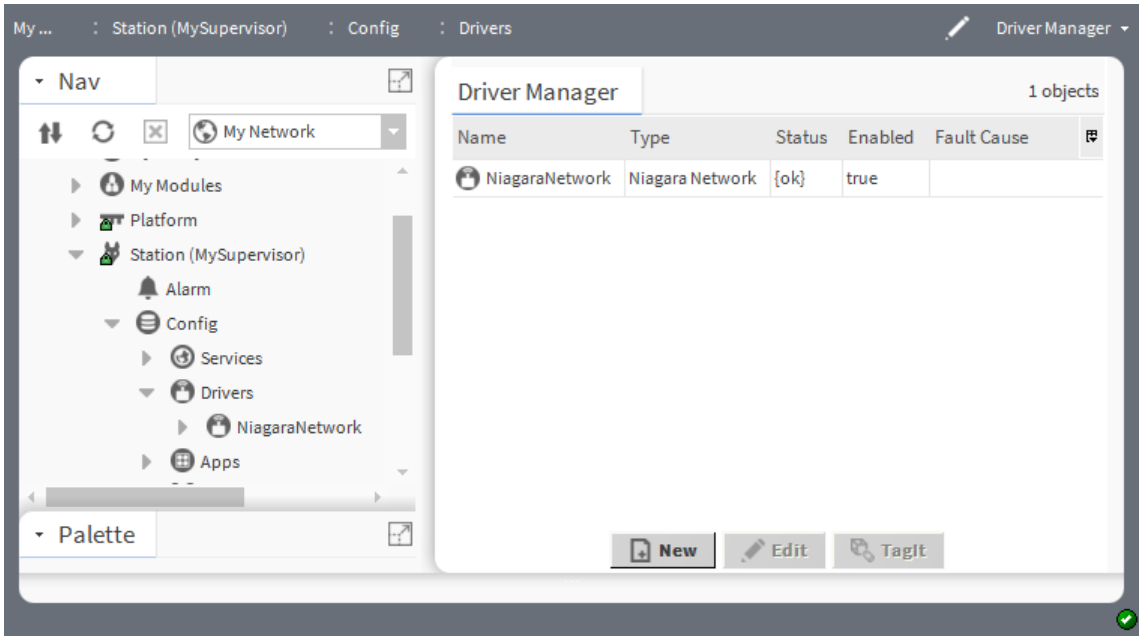
Adding a CcnNetwork using the New button

This procedure uses the **New** button to add a **CcnNetwork** under the station's **Drivers** container.

Prerequisites: You are using Workbench in a PC and are connected to a remote station.

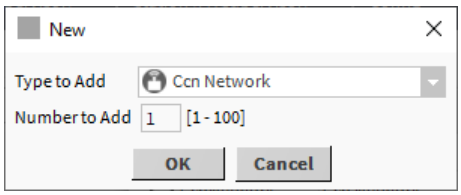
Step 1 To access **CcnNetwork** properties, expand **Config** and double-click the station's **Drivers** container.

The **Driver Manager** opens.



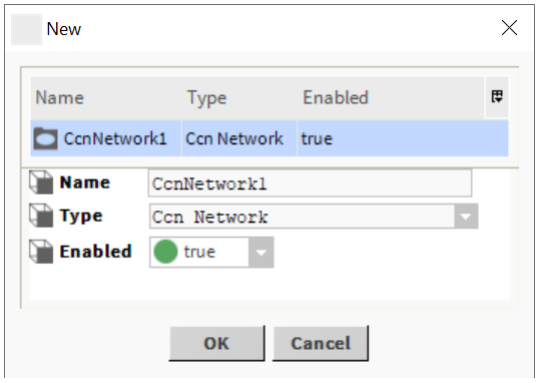
Step 2 Click the **New** button.

The New **Device Network** window opens.



Step 3 Select `CcnNetwork` from the drop-down list, number to add: 1, and click **OK**.

A second **New** window opens.



Step 4 Change the network name or leave it at the default and click **OK**.

The driver adds the network to the database. You should have a network named **CcnNetwork** (or whatever you named it), under your **Drivers** folder.

Step 5 Update at least these properties and click **Save**.

- Bus and element ranges for the: **`First Bus No`**, **`Last Bus No`**, **`Low Elem No`** and **`High Elem No`** for the devices to discover.
- The driver's **`Bus Address`** and **`Elem Address`** (typically 230-238).
- The **`Comm Port`**.

NOTE: On a controller, configure only the available ports.

As with most other drivers, the status of a **CcnNetwork** is either `{Ok}` or less typical `{fault}` (`{fault}` may result from a licensing error). 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.

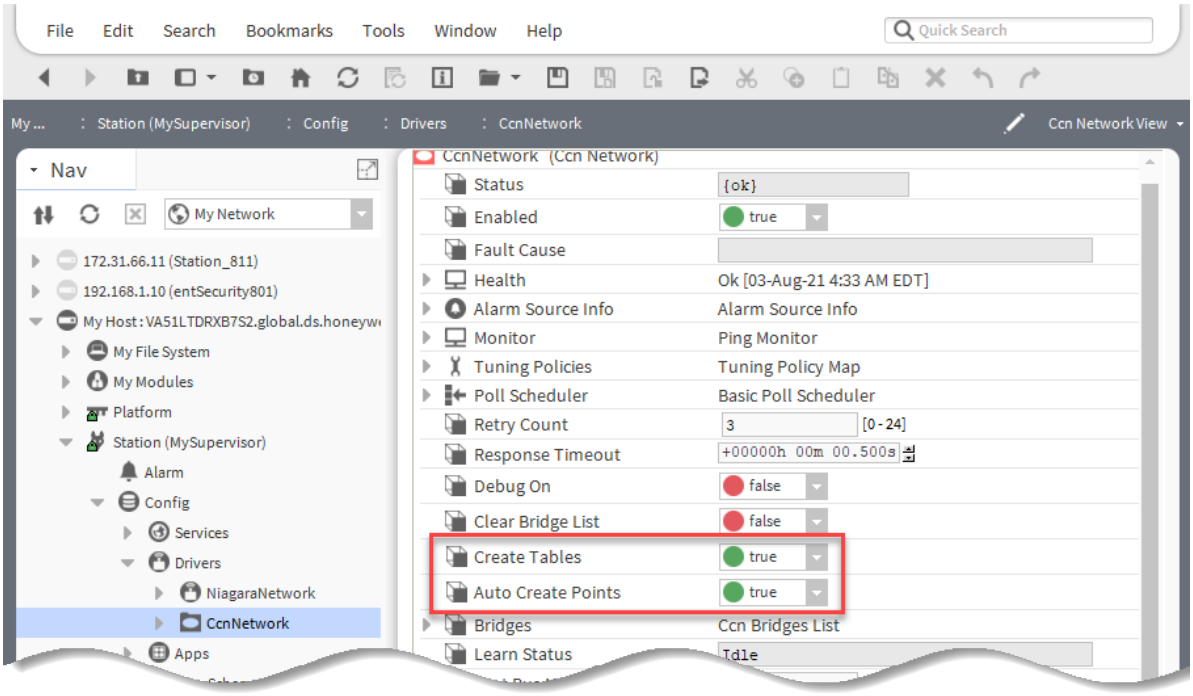
Discovering and adding CcnDevices, tables and points

Discovery can not only locate devices but also automatically add tables and points.

Prerequisites: You added a **CcnNetwork** to the remote station to which you are connected.

Step 1 To access **CcnNetwork** properties, expand **Config→Drivers**, right-click **CcnNetwork** and select **Views→Ccn Network View**.

The **Ccn Network Property Sheet** opens.

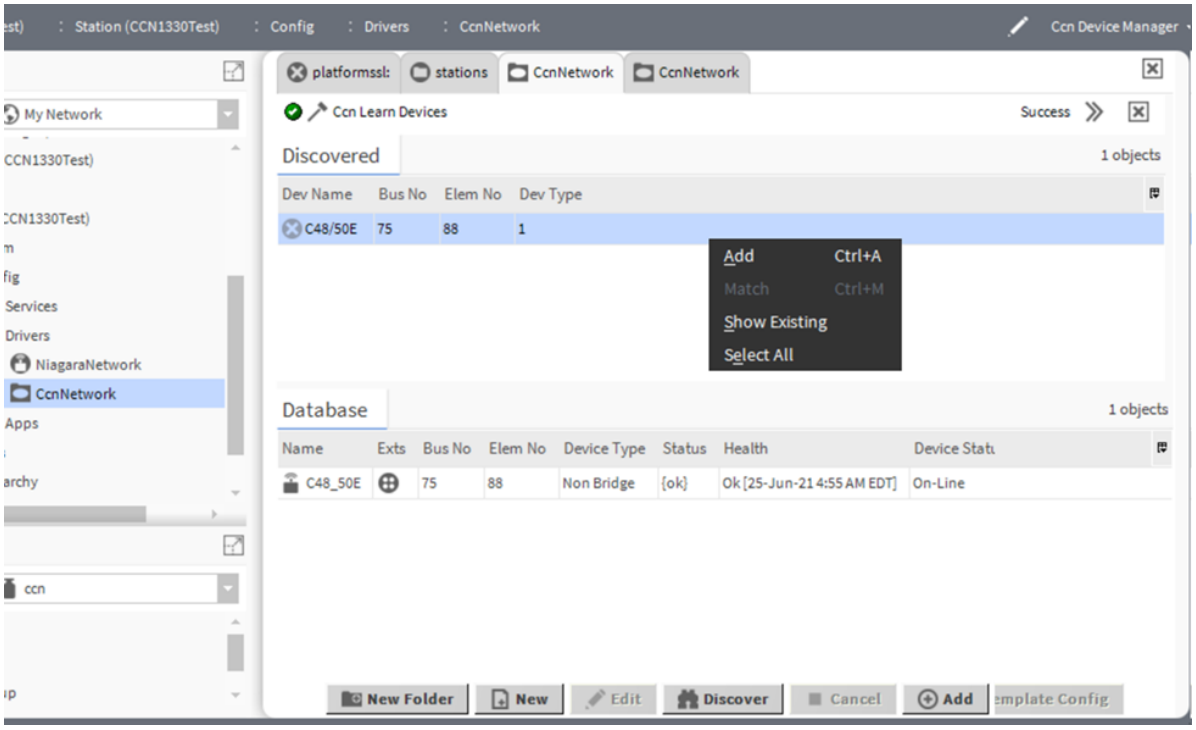


Step 2 To configure a discovery job to discover points, set the following properties to `true`: **`Create Tables`** and **`Auto Create Points`**. Click **Save**.

Enabling these properties ensures that the discovery job finds the points that are under the Status Display and adds them along with the device.

Step 3 In the Nav tree, right-click the **CcnNetwork**. From the menu, select **Views→Ccn Device Manager** and click **Discover**.

The driver discovers the available **CcnDevices**, which are in the given range, and displays them in the **Discovered** pane.



Step 4 Select one or more discovered devices and click **Add**.

When you select device rows in the top **Discovered** pane, the **Add** button becomes active.

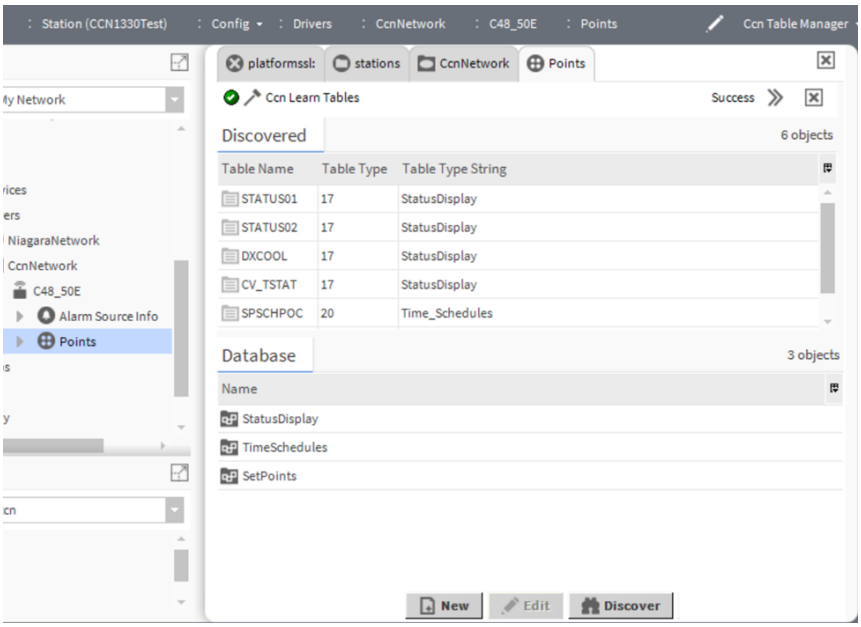
As with other drivers, the CCN driver does not open a window before adding the discovered devices to the station database. It simply adds the entries to database.

The **CcnNetwork**'s monitor routine verifies child **CcnDevices** by pinging them based on duration frequency.

Step 5 To edit **`Device Name`**, **`Bus Num`**, or **`Element Num`**, select a row and click **Edit**.

The device **Edit** window opens.

Step 6 To view the added tables and points, expand **CcnNetwork→CcnDevice→Points** and double-click the type of table.



Step 7 Select the points to add and click **Add**.

The driver adds the discovered table and points to the database.

Adding a CcnDevice from the palette

A **CcnDevice** is the parent for tables and points.

Prerequisites: You added a **CcnNetwork** to the station.

Step 1 To access **CcnNetwork** properties, expand **Config→Drivers** and double-click **CcnNetwork**.

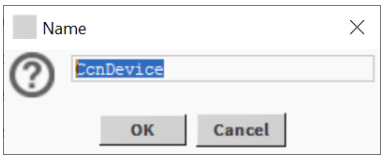
The **Ccn Device Manager** opens.

Step 2 Open the **ccn** palette.

Step 3 Do one of the following:

- Copy the **CcnDevice** component to the **CcnNetwork** node in the Nav tree.
- Drag a **CcnDevice** from the palette to the manager.

The **Name** window opens.

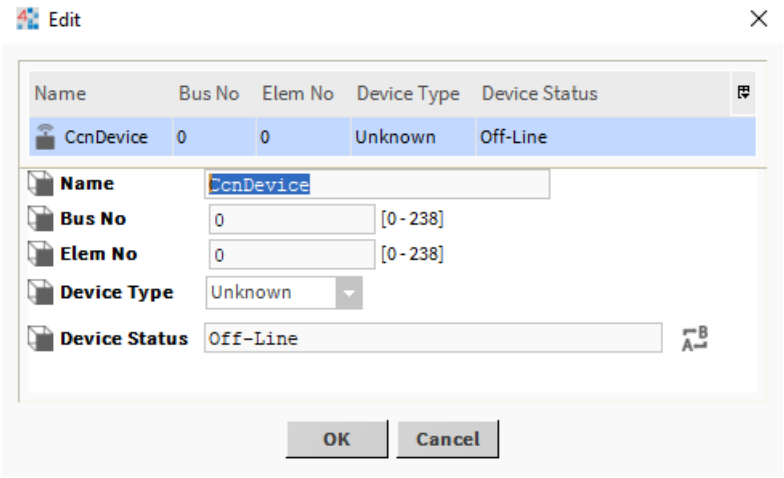


Step 4 Give the device a name or use the default name and click **OK**.

The framework adds the device to the **CcnNetwork**.

Step 5 To configure the **CcnDevice**, double-click the newly created **CcnDevice** in the **Ccn Device Manager**.

The **Edit** window opens.



Step 6 Configure at least **Bus No** and **Elem No** and click **OK**.

Step 7 Right-click the device and click **Actions→Fetch**.

The driver retrieves additional device information, such as pic type, part no, model no, etc.

The **Status** of a **CcnDevice** is either `{ok}` or, less typically, `{fault}`. The **Health** slot contains historical timestamp properties that record the last device transitions from `{ok}` to any other status. The **Fault Cause** property further explains any fault status.

Adding a CcnDevice using the New button

A **CcnDevice** resides under a **CcnNetwork**. This procedure uses the **New** button on the **CCN Device Manager** to add a device.

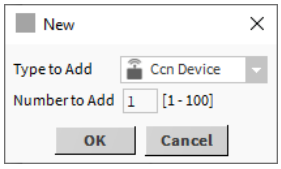
Prerequisites: You added a **CcnNetwork** to the remote station to which you are connected.

Step 1 To access **Ccn Device Manager** expand **Config→Drivers→CcnNetwork** and double-click the **CcnNetwork**.

The **Ccn Device Manager** opens.

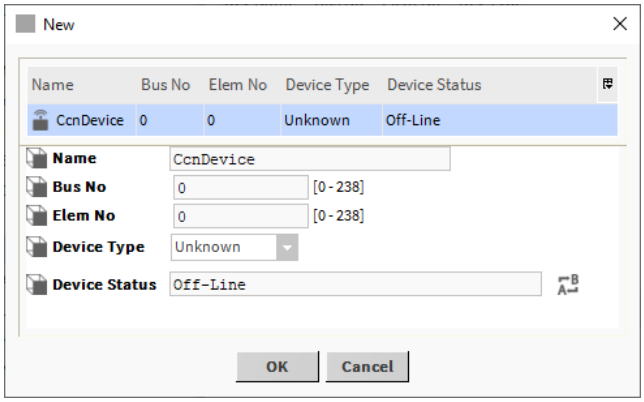
Step 2 Click **New**.

The **New** window for creating a device opens.



Step 3 Select `Ccn Device` and click **OK**.

A second **New** window opens.



Step 4 Configure at least **Bus No** and **Elem No** and click **OK**.

The driver adds the device to the database.

Step 5 Right-click the device and click **Actions→Fetch**.

The driver retrieves additional device information, such as pic type, part no, model no, etc.

The **Status** of a **CcnDevice** is either `{ok}` or, less typically, `{fault}`. The **Health** slot contains historical timestamp properties that record the last device transitions from `{ok}` to any other status. The **Fault Cause** property further explains any fault status.

Designating a station as a CCN alarm or broadcast acknowledger

An acknowledger setting configures the controller to recognize broadcast messages that appear on its CCN bus.

Prerequisites: You are connected to a remote station.

Step 1 To access **CcnNetwork** properties, expand **Config→Drivers** and right-click **CcnNetwork**. From the menu, select **Views→Ccn Network View**.

Step 2 Set the **Alarm Acknowledger** property to `true` and click **Save**.

Designating a station as a CCN time broadcaster

Many CCN devices may broadcast information over the network, such as the time of day or point values like outdoor air temperature. This procedure configures a device to broadcast the time.

Prerequisites: You are connected to the remote station.

Step 1 To access **CcnNetwork** properties, expand **Config→Drivers** and right-click **CcnNetwork**. From the menu, select **Views→Ccn Network View**.

Step 2 Set the **Time Broadcaster** property to `true` and click **Save**.

Step 3 To enable time broadcasting, right-click the **CcnNetwork** and select **Actions→Time Broad Casting**.

Chapter 3 CCN tables and proxy points

Topics covered in this chapter

- ♦ Discovering table groups
- ♦ Configuring a CCN table
- ♦ Updating one point
- ♦ Creating and configuring a CCN object
- ♦ Downloading device tables
- ♦ Uploading device tables
- ♦ Changing to metric units
- ♦ Table Polling

CCN driver points are organized into tables under table groups and displayed in tabular form. Adding a device automatically also adds table groups and proxy points to the database. The collection of CCN proxy point table objects models the tables of a CCN controller.

Proxy points

The collection of CCN proxy point objects model the real and internal I/O as well as selected internal modules of a CCN controller.

Table 1 Proxy points, their behaviour and where they are valid

Type of point	Behaviour	Where it is valid
Ccn Input Proxy	Models a single input point defined by field index and field name in the table. An Input Point is one that is not forceable or writeable.	Valid under CcnPicTable types 11H and 12H (applies to non-forcible points)
Ccn Output Proxy	Models a single forceable output point defined by field index and field name in the table. An Output Point is one that is forceable. You can select the Set Value command.	Valid under CcnPicTable types 11 H and 12H (applies to forcible points)
	or Models a single non-forceable output point defined by field index and field name in the table. This point cannot be forced but can be set.	Valid under: • CcnPicTable types 10H and 13H, or 17H • CcnDataTable types that are under CcnPocTable types 14H, 16H, or 18H
Ccn Fid Output Proxy	Models a single output point defined by field index and field name in the table. An output points is forceable. You can select the Force and Auto commands.	Valid under CcnFidTable types 501H

Points and tables

Table 2 Points and the tables they belong to

Point(s)	Tables
CcnTable proxy point	CcnPicTable, CcnPocTable, CcnDataTable, CcnDataTablewithTimeSchedule, CcnFidTable, CcnFidTablewithTimeSchedule, and CcnAHTable
CcnPicTable or CcnPocTable or CcnAHTable proxy point	can only be added to a **CcnDevice** container generally under a CcnTableGroup

Point(s)	Tables
CcnDataTable or CcnDataTablewithTimeSchedule proxy point	can only be added to a CcnPocTable container generally under a CcnTableGroup
CcnFidTable or CcnFidTablewithTimeSchedule proxy point	can only be added to a **CcnDevice** container generally under a CcnTableGroup

Tables and points

Table 3 Tables and the points they model

CCN table	Points modeled by the table
CcnPicTable	Models a single Pic table defined by table type and table instance.
CcnPocTable	Models a single Poc table defined by table type and table instance.
CcnDataTable	Models a single Data table defined by table type and table instance.
CcnDataTableWithTimeSchedule	Models a single Data table with Time Schedule defined by table type and table instance.
CcnAHTable	Models a single Alarm History table defined by table type and table instance.
CcnFidTable	For IO Points table type 501H, one table entry (instance 1) models all IO point tables (up to 64 points - 1 instance of table type 501H per point) For all other Fid table types supported (Time Schedules, Setpoints, Holidays) , Fid Tables model device tables 1 for 1 by table type and table instance
CcnFidTableWithTimeSchedule	Models a single Fid table with Time Schedule defined by table type and table instance.

Discovering table groups

Independently of adding a device, you may discover table groups and the points they contain.

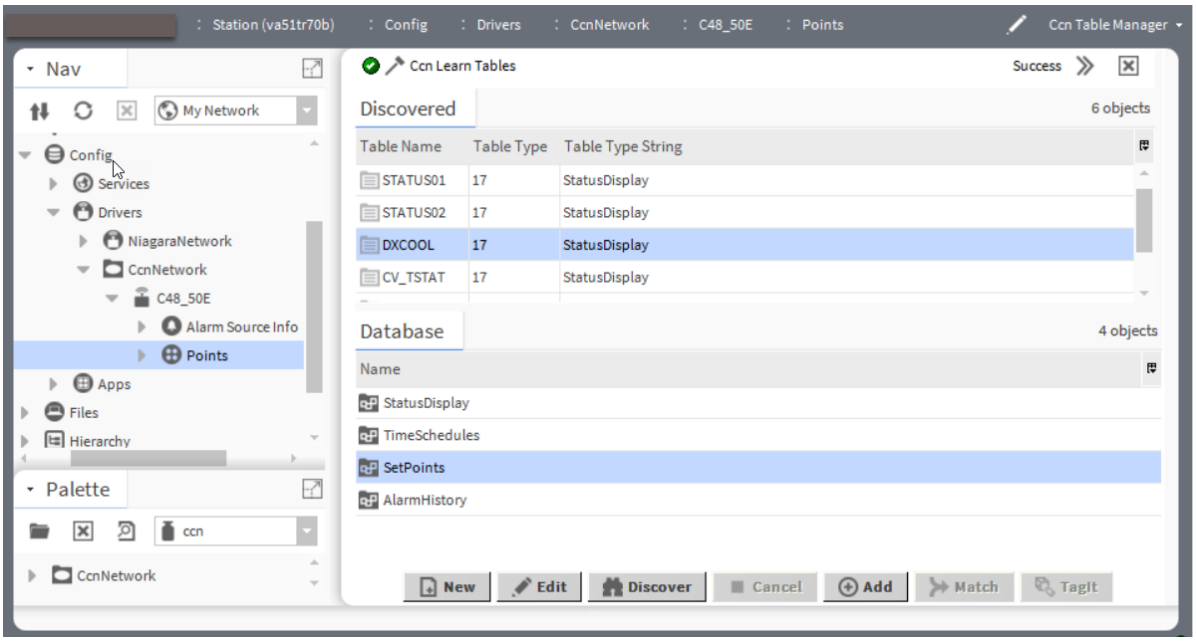
Prerequisites: You created a **CcnDevice** under your **CcnNetwork**.

Step 1 Expand **Config→Drivers→CcnNetwork→CcnDevice** and double-click **Points**.

In the View pane, the **Ccn Table Manager** opens.

Step 2 Click **Discover**.

A discovery job runs and displays the discovered table groups.



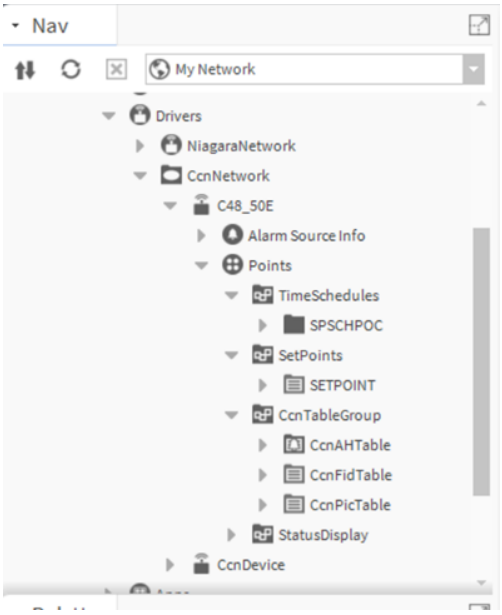
Once the discovery job is complete, the top **Discovered** pane displays the collection of discovered table groups. The success of the job displays in the upper right corner above the **Discovered** pane.

Step 3 To add the table groups to the database, select the table groups and click **Add**.

The **Add** window opens.

Step 4 Select the table groups and click **OK**.

The system adds the discovered table groups to the database and displays them under the **Points** folder by category.



Step 5 To discover the data tables under a currently-discovered table group, such as a POC table, double-click the table group.

Step 6 In the view pane, click **Discover**.

A discovery job runs and displays the results.

Step 7 To add the discovered tables to the database, select the tables.

Step 8 Click **Add**.

Configuring a CCN table

Most conveniently you can add a **CcnTable** during the discovery process of **CcnDevices**. Alternatively, to add a **CcnTable** to an existing station, proceed as described in the following procedure.

Prerequisites: The parent table group exists.

Step 1 To open a **Ccn Table Manager**, expand **Config→Drivers→CcnNetwork→CcnDevice** and double-click the **Points** folder.

The **Ccn Table Manager** opens.

Step 2 To add a table to the group, click **New**. Select the type of table and click **OK**.

This adds the **CcnTable** to the CCN device.

Step 3 Double-click the table. Set the **`Table Type`** and **`Table Instance`** properties to their actual values in decimal, not hex and click **Save**.

If the table type is 501H (for Fid IO points), set the instance to 1.

If the table type is **CcnDataTable** or **CcnDataTableWithTimeSchedule**, also set the **`Block Number`**.

Step 4 Do one of the following:

- Click **Commands→Fetch** (menu bar under command).
- At the bottom of the view, click **Fetch**.

The driver retrieves additional table information and builds it to the point list.

NOTE: The table name “SPSCHPOC” does not support the manual addition of a table and fetching.

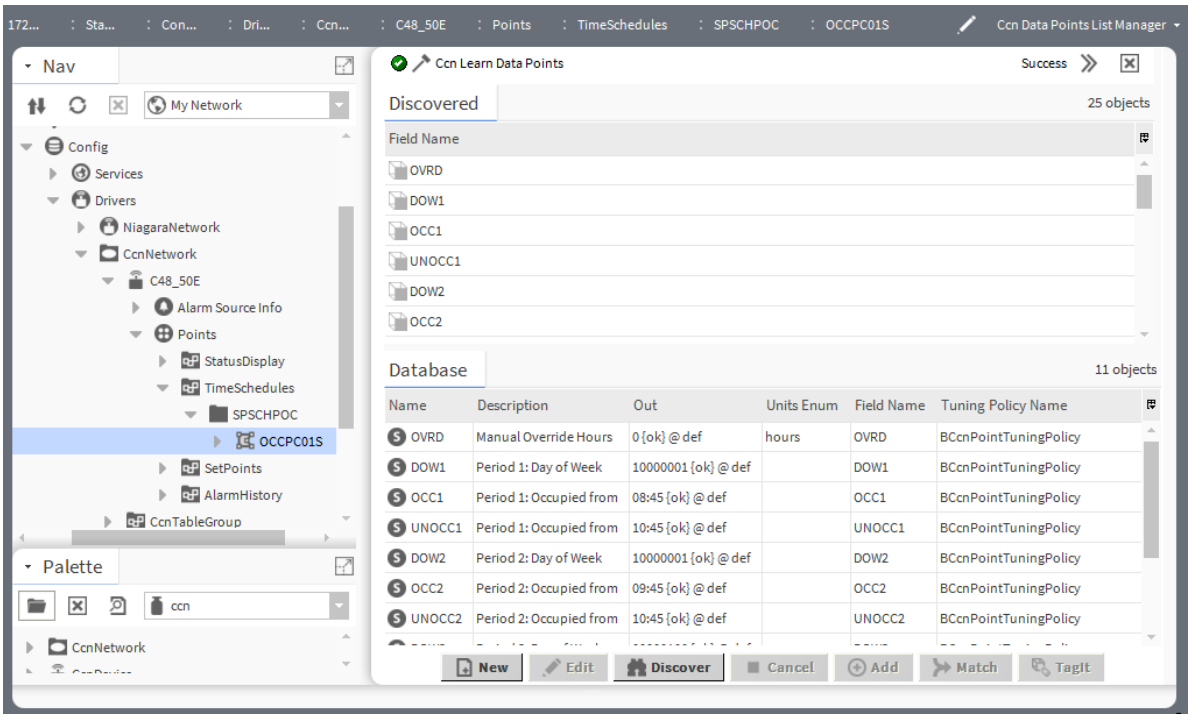
Updating one point

Tables organize proxy points into groups.

Step 1 To view the **Ccn Data Points List Manager**, in the Nav tree expand **Config→Drivers→CcnNetwork→CcnDevice →Points**. Under **Points** a list of table groups appears by category. Right-click a table group and select **Views→Ccn Data Points List Manager** from the menu.

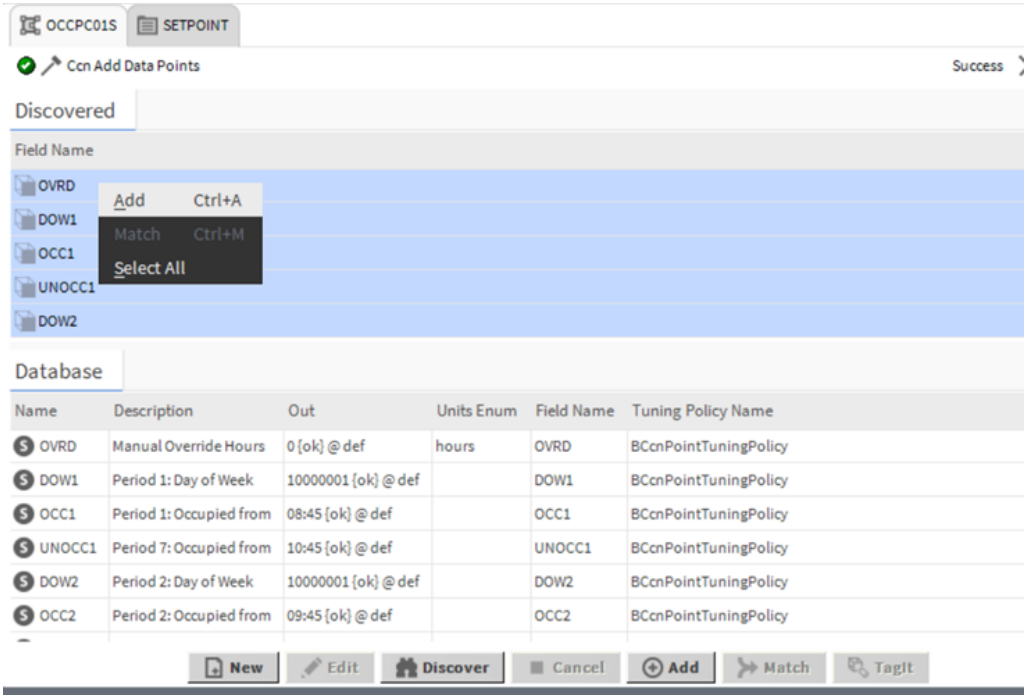
Step 2 In the **view** pane, click **Discover**.

A discovery job runs and displays the discovered points in the **Discovered** pane.



Step 3 To add the discovered tables, select and right-click the selected tables.

A menu opens.



Step 4 Select `Add` or press **Ctrl+A**.

The system adds the discovered points to the **Database**.

Step 5 In the Ccn **Nav** tree, expand **Config→Drivers→CcnNetwork→CcnDevice →Points**. Under **Points** double-click the Ccn data table.

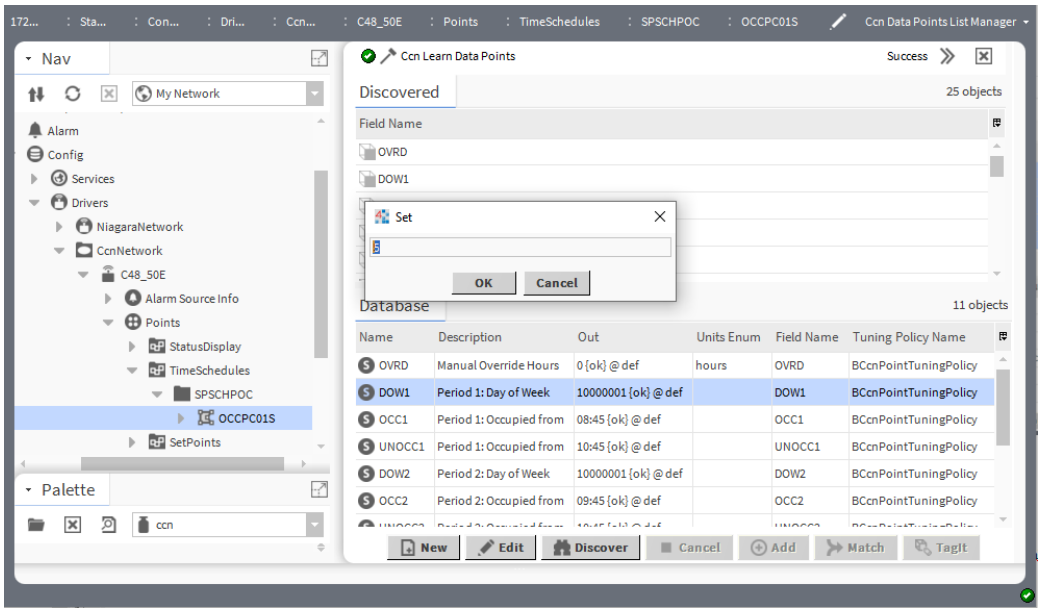
In the **view** pane, the **Ccn Data Table Manager** opens.

Step 6 To discover points, click **Discover** in the **view** pane of the **Ccn Data Table Manager**.

The **Discovered** pane opens displaying available points.

Step 7 To change a point value, right-click a selected point and select **Actions→Set** from the menu.

The **Set** window opens.



The screen capture above shows a digital value being set.

Step 8 Enter the value and click **OK**.

The point value changes accordingly.

Creating and configuring a CCN object

You can only add a CCN object under a **CcnPicTable** or **CcnDataTable** using the **Ccn Fid Table Point List Manager**.

Set Command, **Override Command**, and **Auto Command** are actions that you can find on the**CcnObject** depending on its type. The **Set command** writes a non-forcible Ccn object.

Step 1 To perform this command, add the **CcnObject** to the station.

Step 2 Right-click **CcnObject**. From the menu, select **Actions→Set command**. You can issue this action using the **Point List Manager** or **Wire Sheet** view.

If you set the value within the limits, the driver writes the new value to the field device.

Step 3 The **Override command** forces a Ccn Output object on if it is a discrete point type with a value of 1, and an active text as defined by the one's value of the Discrete Text Offset text pair. This command forces the object off if it has no value (0) with an inactive text as defined by the no value (0) of the Discrete Text Offset text pair. The CCN driver manages two force levels (level 4 is used for commands, level 8 is used for links) and the auto level (0). The command level force is the highest priority available from the driver.

If the CcnOutput object is a discrete point type, the command opens a combo box, with `Force On (Discrete on Text)` and `Force Off (Discrete off Text)` options.

Step 4 If you select either of these commands, a Force command will be issued. After the Force command is successfully completed, the value field(s) of the Point Entry and CcnOutput object reflect the new value and the row in the **Point List Manager** view changes to the color lavender.

Step 5 To perform a point value Force command, add the Ccn object to the station database.

Step 6 Right-click the Ccn object. From the menu, select **Actions→Override command.**

You can override a value using the **Point List Manager** or **Wire Sheet** view. Auto command removes the force level command on a CcnOutput object.

After the Auto Command is successfully completed, the value field(s) of the Point Entry and CcnOutput object reflect a new value.

Step 7 To perform this command, add the CcnObject to the database.

Step 8 Right-click the **CcnObject**. From the menu, select **Actions→Auto command**.

You can override a value using the **Point List Manager** or **Wire Sheet** view.

Downloading device tables

Downloading moves data from the station to the CCN devices. A **CcnDevice**'s **CcnPicTables**, **CcnPocTables**, **CcnDataTables** and **CcnFidTables** can be downloaded.

Prerequisites: You added a **CcnNetwork** to the station. The **ccn** palette is open.

Step 1 Expand **CcnNetwork**.

Step 2 To download tables, right-click **CcnDevice** and click **Actions→Download**.

The **Download** window opens.

Step 3 To continue, click **OK**.

Uploading device tables

Uploading moves data from CCN devices to the station. Upload the **CcnDevice**'s **CcnPicTables**, **CcnPocTables**, **CcnDataTables** and **CcnFidTables**.

Prerequisites: You are connected to the remote station.

Step 1 Expand **CcnNetwork→CcnDevices**. Drag **CcnTableGroup** to **CcnNetwork**.

Step 2 To upload the devices, right-click **CcnDevices**. From the menu, select **Actions→Upload→OK**.

Changing to metric units

The CCN driver defaults to imperial units. This procedure explains how to change to metric units.

Prerequisites: You added **CcnNetwork** to the station.

Step 1 Expand **Config→Drivers** and double-click the **CcnNetwork**.

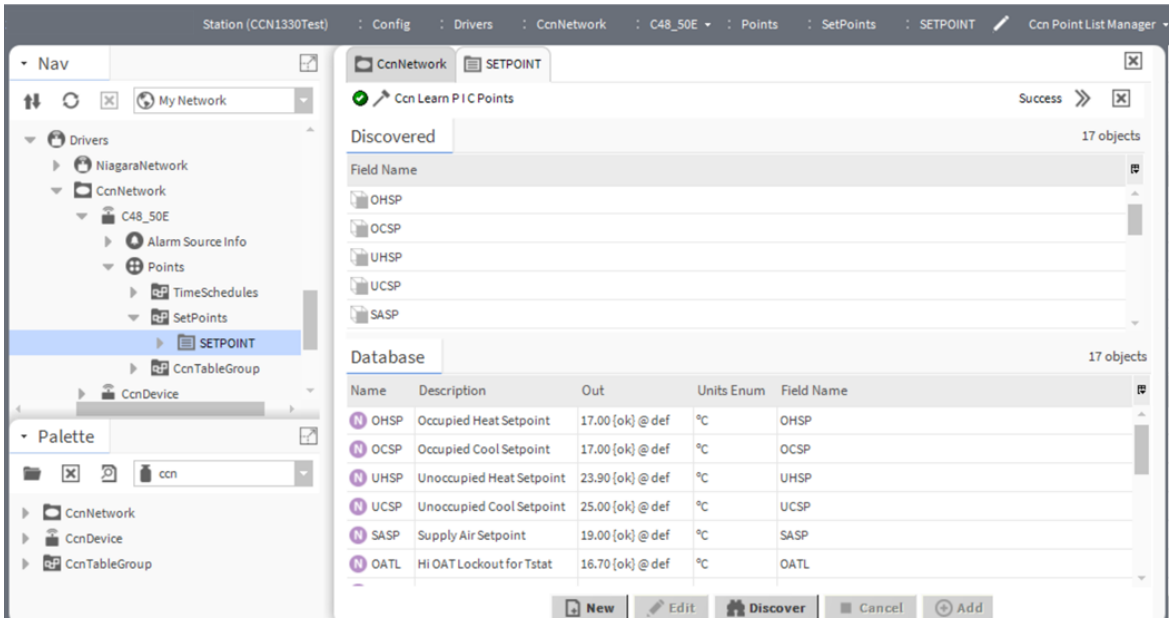
The **Ccn Device Manager** opens.

Step 2 In the View pane, click the **View Selector** and open the **Property Sheet** view.

Step 3 Enable (`true`) or disable (`false`) for the **`Display Metric`** property.

Step 4 To open **Ccn Point List Manager** view, expand **Points→SetPoints→SETPOINT**. Double-click **SETPOINTS**.

Step 5 In the View pane, click **Discover**.



The screen capture is an example of the **Ccn Point List Manager** when **`Display Metric`** is enabled (`true`).

When you change the **`Display Metric`** property, the units change accordingly.

Step 6 To update units, refresh Workbench.

Table Polling

The CCN driver polls connected devices at the table level.

CCN **CcnPicTable** and **CcnDataTable** objects poll when they are in view based on **`Poll Frequency`**. The default frequency is the `Normal`. You can change the duration of this frequency.

Chapter 4 Components

Topics covered in this chapter

- ccn-CcnNetwork
- ccn-CcnBridgesList
- ccn-CcnUnsolicitedReceive
- ccn-CcnDevice
- ccn-CcnTableGroup
- ccn-CcnAHTable
- ccn-CcnFidTable
- ccn-CcnPicTable
- ccn-CcnPocTable
- ccn-CcnDataTable
- ccn-CcnDataTableWithTimeSchedule
- Ccn-CcnObject

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:

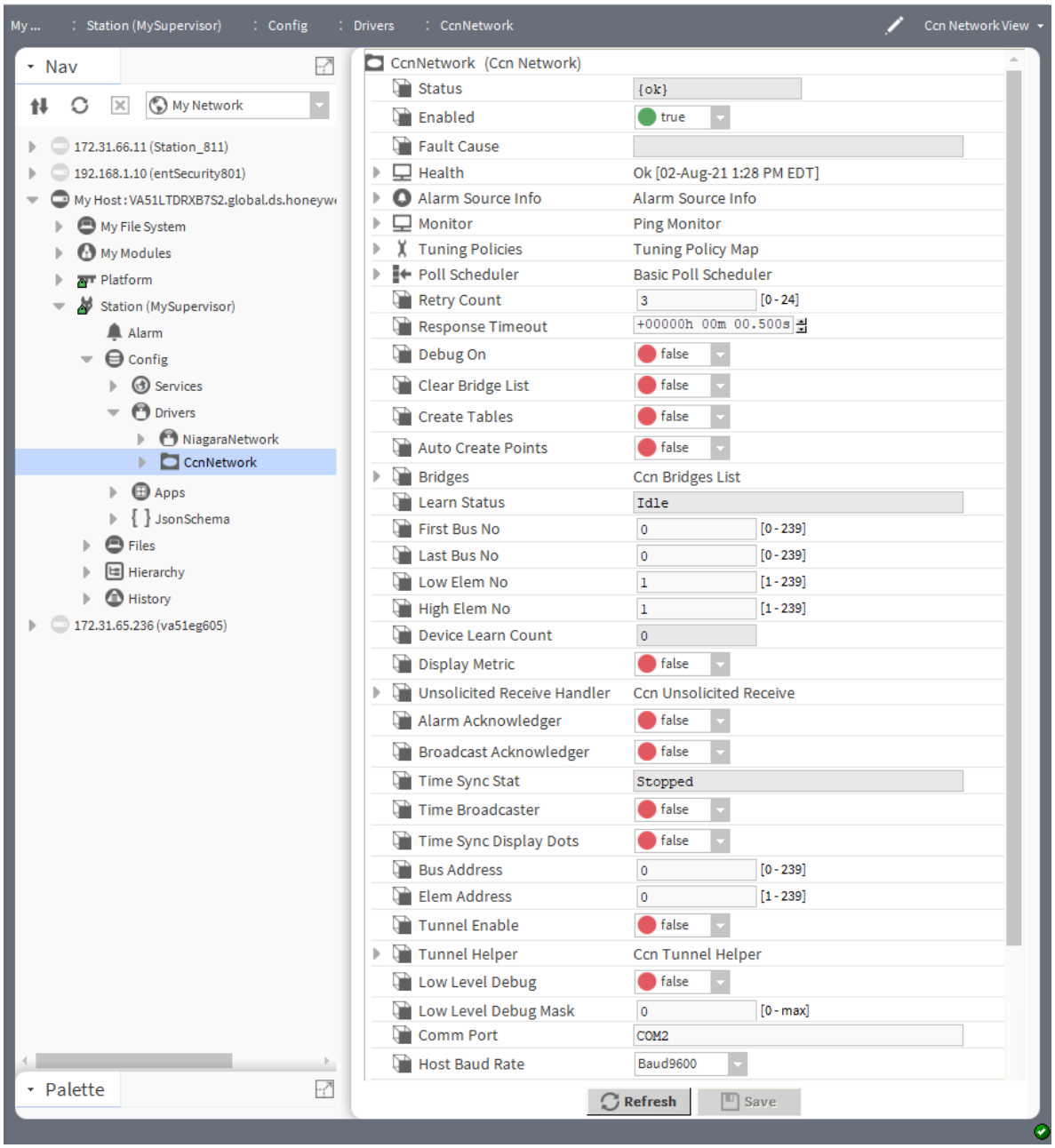
- Right-clicking on the object and selecting **Views→Guide Help**
- Clicking **Help→Guide On Target**

ccn-CcnNetwork

This component is a container object used to track the status of the entire CCN, track and perform time synchronization between the CCN system and the Niagara system, and provide support for automatically creating proxy points by learning (discovering) devices or controllers within the CCN.

CcnNetwork is the top-level CCN container component in a station.

Figure 2 CcnNetwork properties



To access this view, expand **Config→Drivers**, right–click **CcnNetwork** and click **Views→Ccn Network View**.

In addition to the standard driver properties (Status, Enabled, Fault Cause, Health, Alarm Source Info, Monitor, Tuning Policies and Poll Scheduler), these properties are unique to this component.

Property	Value	Description
Response Timeout (sec)	hours, minutes, seconds (recommended 2-5 seconds)	Indicates how long the driver waits for a response before declaring the CCN non-responding. For a poll/response sequence, if the response does not return within this period, the driver attempts a retry. If **`Retry Count`** has been exhausted, the driver declares a communications failure.
Clear Bridge List	`true` or `false` (default)	Clears the **`Bridges`** list during a discovery and starts over.
Create Tables	`true` or `false` (default)	Automatically creates (`true`) all tables under all devices that have been selected for **CcnDevice** creation during the create process.
Auto Create Points	`true` or `false` (default)	Enables (`true`) and disables (`false`) automatic point creation when the driver creates Pic and or Data and or FID IO tables. This property only supports the creation of points under Status Display (11H) and FID Status tables (501H).
Bridges	read-only	Reports the known bridge addresses.
Learn Status	read-only	Reports the status of the network-level learn command (`busy`, `idle` or `error`).
First Bus No	number	Defines the starting bus address to discover.
Last Bus No	number	Defines the ending bus address to be used by discovery.
Low Elem No	number	Defines the starting element address to be used by discovery.
High Elem No	number	Defines the ending element address to be used by discovery.
Device Learn Count	number	Reports the number of devices found during the latest discovery.
Display Metric	`true` or `false` (default)	Selects metric (`true`) or imperial (`false`) units for of all values.
Unsolicited Receive Handler	read-only	Reports unsolicited messages.
Alarm Acknowledger	`true` or `false` (default)	Enables (`true`) and disables (`false`) the use of a controller to acknowledge alarms.
Broadcast Acknowledger	`true` or `false` (default)	Enables (`true`) and disables (`false`) the use of a controller to broadcast acknowledgments.
Time Sync Stat	read-only	Reports if the time sync service has started or stopped.
Time Broadcaster	`true` or `false` (default)	Enables (`true`) and disables (`false`) the use of the **CcnNetwork** as a time broadcaster.
Time Sync Display Dots	`true` or `false` (default)	Enables (`true`) and disables (`false` a “T” character display in the diagnostic output for every Time Sync cycle.
Bus Address	number	Defines the CCN bus address that the station is connected to (generally the primary bus, bus 0).

Property	Value	Description
Elem Address	number	Defines the CCN element address on the **Bus Address** that the station is assigned to. This is generally a **High element No** just below the broadcast address range, typically 230-239. Do not use the same address assigned to the ComfortVIEW application that you use to tunnel for setup and configuration.
Low Level Debug	`true` or `false` (default)	Turns low-level debug messages on and off. `true` configures a controller to be the CCN network broadcast acknowledger causing the driver to generate specific debug text from low-level native code and send it to the **administrator console** window. `false` disables this feature.
Low Level Debug Mask	number (defaults to zero (0))	Do not use this property to isolate a specific problem without the assistance of Tridium Engineering.
Comm Port	communications port number	Defines the interfaces between a computer and other computers or devices.
Host Baud Rate	drop-down list (defaults to `Baud9600`)	Defines communication speed in bits per second.

ccn-CcnBridgesList

This component serves as a container for CCN bridges.

To access this sub-component on the **CcnNetworkProperty Sheet**, expand **Config→Drivers**, right-click **CcnNetwork**, click **Views→Ccn Network View** and double-click **Bridges**.

This component lists the bridges. It contains no properties to configure.

ccn-CcnUnsolicitedReceive

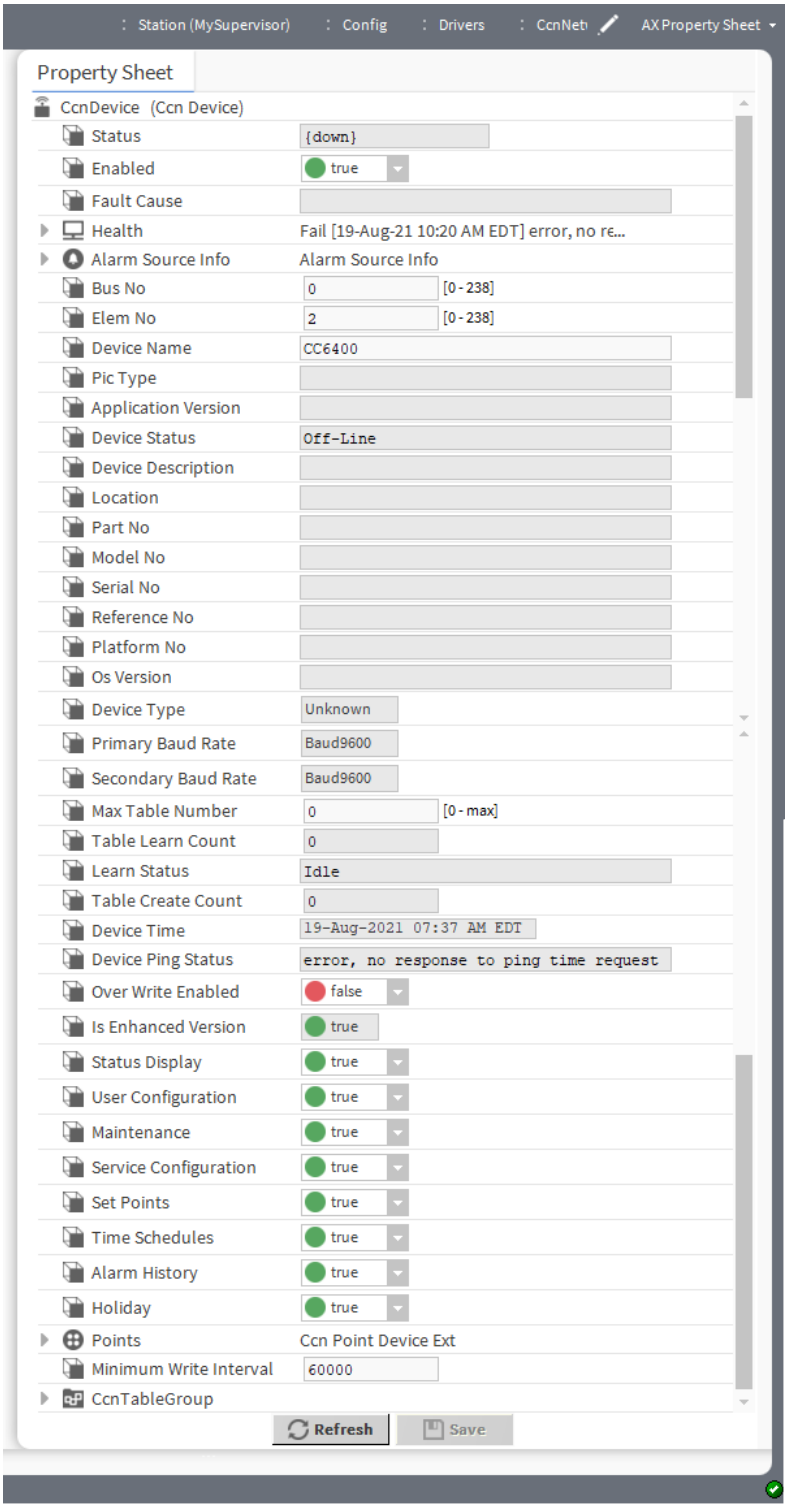
This component provides access to a single read-only property, **Unsolicited Message Count**. This property reports the number of unsolicited messages received.

To access this component, expand **Config→Drivers**, right-click **CcnNetwork**, click **Views→Ccn Network View** and double-click **Unsolicited Receive Handler**.

ccn-CcnDevice

This component is a container object used to track the status of each CCN device, track and perform time synchronization between a CCN device and the system and provide support for automatically creating proxy points using tables within the CCN device.

Figure 3 CcnDevice properties



To access these properties, expand **Config→Drivers→CcnNetwork** and double-click **CcnDevice**.

In addition to the standard driver properties (Status, Enabled, Health, Fault Cause), these properties are unique to this component.

Property	Value	Description
Bus No	number	Defines the network bus on which the device is installed.
Elem No	number	Defines the element address of the device.
Device Name	text	Sets up the device name.
Pic Type	read-only	Reports the Pic Type retrieved from the device with the fetch command.
Application Version	read-only	Reports the app version retrieved from the device with the fetch command.
Device Status	read-only	Reports the current condition of the device (`online` or `offline`).
Device Description	read-only	Reports the device description retrieved from the device with the fetch command.
Location		Reports the device location retrieved from the device with the fetch command.
Part No	read-only	Reports the device part number retrieved with the fetch command.
Model No	read-only	Reports the device model number retrieved from the device with the fetch command.
Serial No		Reports the device serial number retrieved from the device with the fetch command.
Reference No	read-only	Reports a number retrieved from the device with the fetch command.
Platform No	read-only	Reports a number retrieved from the device with the fetch command.
Os Version	read-only	Reports the current version of the operating system as retrieved from the device with the fetch command.
Device Type	read-only	Reports the type of device (`Bridge` or `NonBridge`).
Primary Baud Rate	read-only	Reports the baud rate as retrieved from the device with the fetch command.
Secondary Baud Rate	read-only	Reports a second baud rate retrieved from the device with the fetch command.
Max Table Number	number	Identifies the device using a number. This is beneficial, because it shortens the discovery job. Some devices do not support the technique used to automatically learn the **`Max Table Number`**, so you may enter this value if you know it.
Table Learn Count	read-only	Reports the number of tables discovered during latest discovery job.
Learn Status	read-only	Reports the status of the network-level learn command (`busy`, `idle` or `error`).
Table Create Count	read-only	Reports the number of tables created during the latest create process.
Debug On	`true` or `false` (default)	Turns debugging on (`true`) and off (`false`).

Property	Value	Description
Debug Discovery On	`true` or `false` (default)	Turns debugging during a discovery job on (`true`) and off (`false`).
Device Time	read-only	Reports the latest device date and time as returned in response to a query of the device's Date Time Table sent by the ping process. Device date and time are not used by Niagara for any purpose other than a short and quick message to perform a device status check, so if they are not current and the next property (Device Ping Status) indicates "skipped, not needed as the last ping communicated the child object." This is a good sign in that adequate successful comm activity is occurring and the Device Ping that updates Device Time is not required to run.
Device Ping Status	read-only	Reports the success or failure status of the device ping. The ping process alternately retrieves the date and the time block. A success message would be: "succeeded and parsed date from ping message" "succeeded and parsed time from ping message" "skipped, not needed since child object communicated since last ping" "received date response but with NAK" "received time response but with NAK" ** The NAK response means the specific device does not maintain date and/or time data. Failure messages would be: "skipped, device is out of service" "failed, no response to date request" "failed, no response to time request" "could not complete last ping:"
Ccn Device Enabled	`true` or `false` (default)	Turns the connection to the device on (`true`) and off (`false`).
Is Enhanced Version	read-only	Indicates if the controller license designates this device as enhanced or not. If enhanced, access to configuration table data is supported. If not enhanced, access is limited to Display, Setpoint, and Time Schedule Data.
Status Display	`true` or `false` (default)	Enables (`true`) and disables (`false`) the status display.
User Configuration	`true` or `false` (default)	Enables (`true`) and disables (`false`) the use of configuration settings to discover proxy points.
Maintenance	`true` or `false` (default)	Enables (`true`) and disables (`false`) the use of maintenance information in the point discovery job.

Property	Value	Description
Service Configuration	`true` or `false` (default)	Enables (`true`) and disables (`false`) the use of service information in the point discovery job.
Set Points	`true` or `false` (default)	Enables (`true`) and disables (`false`) the use of point values in the discovery job.
Time Schedules	`true` or `false` (default)	Enables (`true`) and disables (`false`) the use of time schedules in the point discovery job.
Alarm History	`true` or `false` (default)	Enables (`true`) and disables (`false`) the use of alarm conditions in the point discovery job.
Holiday	`true` or `false` (default)	Enables (`true`) and disables (`false`) the use of holidays in the point discovery job.

CcnDevice Actions

Ping accesses the CCN device and updates device the ping status properties.

Upload uploads a list of CcnTables blocks from a CcnDevice's, CcnPicTables, CcnPocTables, CcnDataTables and CcnFidTables and updates all Station resident data to match the data retrieved from the field device.

Download constructs each CCN device's non-real-time table's value blocks from the station-resident data into CcnPicTables, CcnPocTables, CcnDataTables and CcnFidTables and downloads the tables to the field device.

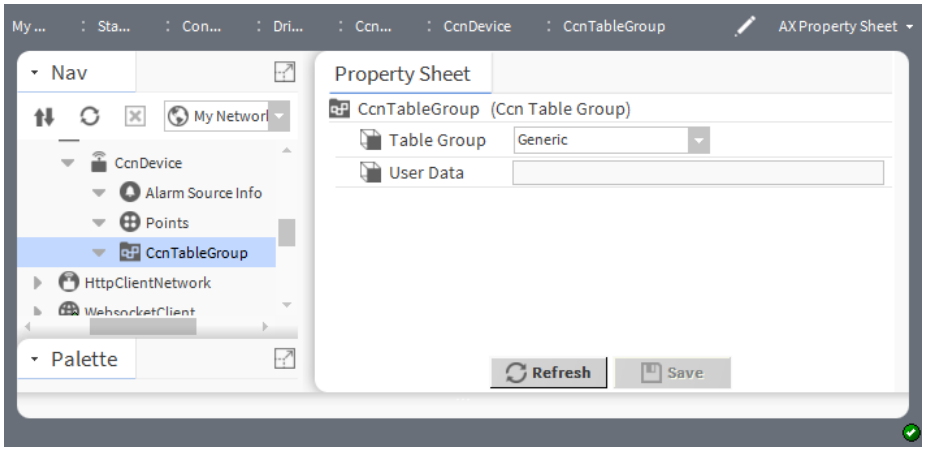
NOTE: Downloading tables to a controller has not been tested.

Fetch retrieves additional information from the device when you manually add a new device with the correct bus and element numbers.

ccn-CcnTableGroup

This component sets up a table in the device. The driver uses tables to organize and group proxy points. There are three types of tables: PIC, POC and Data. The table types share some properties, but each has its own unique properties.

Figure 4 Ccn Table Group properties

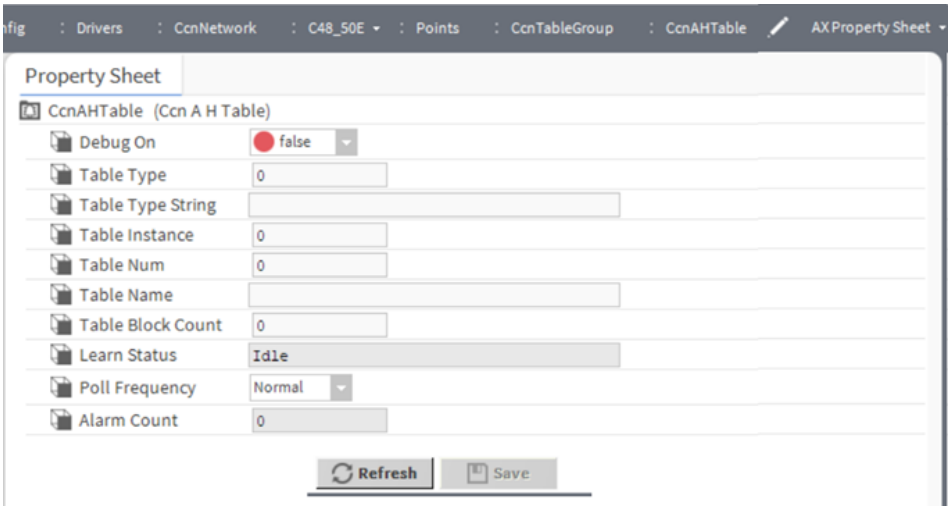


To access these properties, expand **Config→Drivers→CcnNetwork→CcnDevice**, right-click **CcnTableGroup** and click **Views→AX Property Sheet**.

Property	Value	Description
Table Group	drop-down list	Selects the type of group.
User Data	text	

ccn-CcnAHTable

This table displays points for your Carrier air handling units.



To access these properties, expand **Config→Drivers→CcnNetwork→CcnDevice**, double-click **CcnTableGroup** and double-click a **Ccn A H Table**.

Property	Value	Description
Debug On	`true` or `false` default	Turns debugging on (`true`) and off (`false`).
Table Type	number	Configures the type of table. For AH tables, valid type is 67 {43H}.
Table Type String	text	Describes the table.
Table Instance	number	Assigns a table number to the table.
Table Num	number	Assigns a unique table number within **`Table Instance`**.
Table Name	text	Assigns an eight-character name to the table. This is the only property that goes to field device. To change the name in the device after entering it here, invoke the **Fetch** action on the table. An invalid table name is treated as "T." The driver sends the first eight characters of the name to the field device. **NOTE:** Change the **`Table Name`** property on the **Property Sheet** only. Avoid renaming the table name on the wire sheet, slot sheet etc. This operation is controller specific.
Table Block Count	number	Configures the table's block count (generally 10-15).

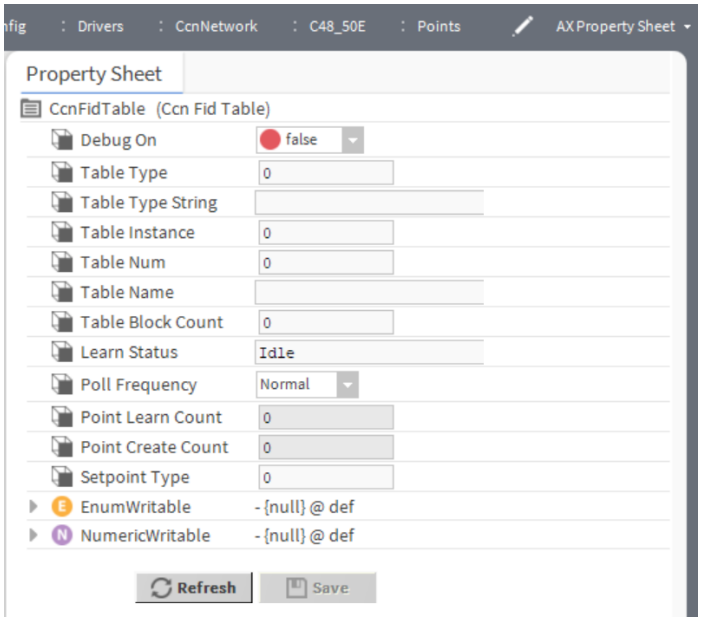
Property	Value	Description
Learn Status	read-only	Reports the status of the network-level learn command (`busy`, `idle` or `error`).
Poll Frequency	drop-down list	Selects one of three pre-configured polling frequencies: `Slow`, `Normal`, `Fast`.
Alarm Count	read-only	Reports the number of current alarms for this table.

Actions

- **Fetch** retrieves additional table-related information.

ccn-CcnFidTable

This table configures data for FID-type proxy points.



To access these properties, expand **Config→Drivers→CcnNetwork→CcnDevice**, double-click **CcnTableGroup** and double-click a **CcnFidTable**.

Property	Value	Description
Debug On	`true` or `false` default	Turns debugging on (`true`) and off (`false`).
Table Type	number	Configures the type of table. For FID tables, valid type is 501H.
Table Type String	text	Describes the table.
Table Instance	number	Assigns a table number to the table.
Table Num	number	Assigns a unique table number within **`Table Instance`**.
Table Name	text	Assigns an eight-character name to the table. This is the only property that goes to field device. To change the name in the device after entering it here, invoke the **Fetch** action on the table.

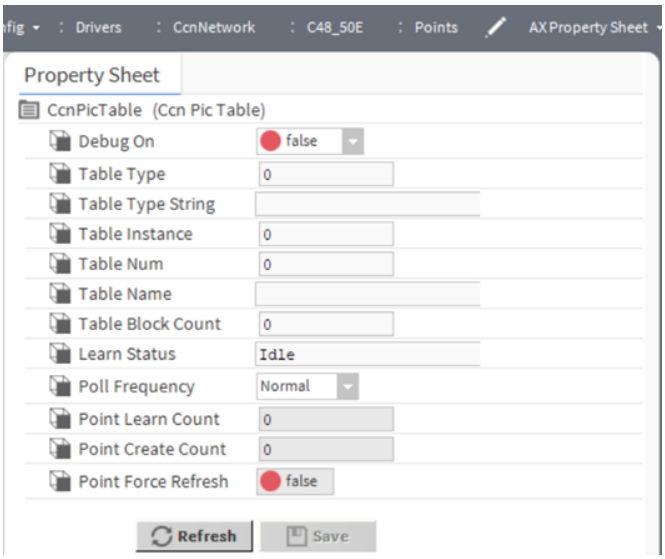
Property	Value	Description
		An invalid table name is treated as "T." The driver sends the first eight characters of the name to the field device. **NOTE:** Change the **`Table Name`** property on the **Property Sheet** only. Avoid renaming the table name on the wire sheet, slot sheet etc. This operation is controller specific.
Table Block Count	number	Configures the table's block count (generally 10-15).
Learn Status	read-only	Reports the status of the network-level learn command (`busy`, `idle` or `error`).
Poll Frequency	drop-down list	Selects one of three pre-configured polling frequencies: `Slow`, `Normal`, `Fast`.
Point Learn Count	number	Reports the number of points found during the latest discovery job.
Point Create Count	number	Reports the number of points created during the latest create process.
Setpoint Type	number	Reports the type of setpoint.

Actions

- **Fetch** retrieves additional table-related information.

ccn-CcnPicTable

This table contains CCN PIC-type proxy points.



To access these properties, expand **Config→Drivers→CcnNetwork→CcnDevice**, double-click **CcnTableGroup** and double-click a **CcnPicTable**.

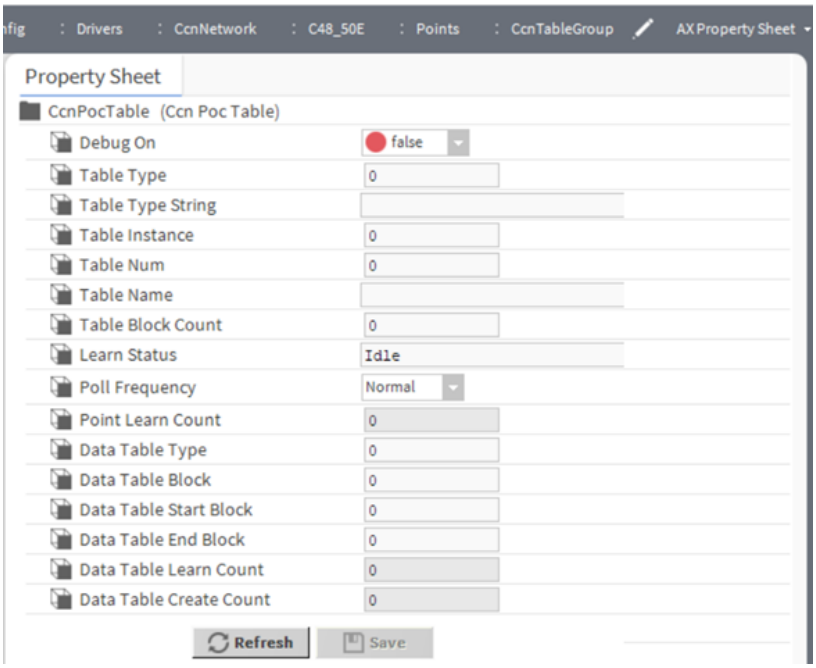
Property	Value	Description
Debug On	`true` or `false` default	Turns debugging on (`true`) and off (`false`).
Table Type	number	Configures the type of table. Valid types are 16 {10H}, 17 {11H}, 18 {12H}, 19 {13H} and 23 {17H}.
Table Type String	text	Describes the table.
Table Instance	number	Assigns a table number to the table.
Table Num	number	Assigns a unique table number within **`Table Instance`**.
Table Name	text	Assigns an eight-character name to the table. This is the only property that goes to field device. To change the name in the device after entering it here, invoke the **Fetch** action on the table. An invalid table name is treated as "T." The driver sends the first eight characters of the name to the field device. **NOTE:** Change the **`Table Name`** property on the **Property Sheet** only. Avoid renaming the table name on the wire sheet, slot sheet etc. This operation is controller specific.
Table Block Count	number	Configures the table's block count (generally 10-15).
Learn Status	read-only	Reports the status of the network-level learn command (`busy`, `idle` or `error`).
Poll Frequency	drop-down list	Selects one of three pre-configured polling frequencies: `Slow`, `Normal`, `Fast`.
Point Learn Count	number	Reports the number of points found during the latest discovery job.
Point Create Count	number	Reports the number of points created during the latest create process.
Point Force Refresh	`true` or `false` default	Turns this function on (`true`) and off (`false`).

Actions

- **Fetch** retrieves additional table-related information.

ccn-CcnPocTable

This table configures data for POC-type proxy points.



To access these properties, expand **Config→Drivers→CcnNetwork→CcnDevice**, double-click **CcnTableGroup** and double-click a **CcnPocTable**.

Property	Value	Description
Debug On	`true` or `false` default	Turns debugging on (`true`) and off (`false`).
Table Type	number	Configures the type of table. Valid types are 20 {14H}, 21 {15H}, 22 {16H} and 24{18H}.
Table Type String	text	Describes the table.
Table Instance	number	Assigns a table number to the table.
Table Number	number	Assigns a unique table number within **`Table Instance`**.
Table Name	text	Assigns an eight-character name to the table. This is the only property that goes to field device. To change the name in the device after entering it here, invoke the **Fetch** action on the table. An invalid table name is treated as "T." The driver sends the first eight characters of the name to the field device. **NOTE:** Change the **`Table Name`** property on the **Property Sheet** only. Avoid renaming the table name on the wire sheet, slot sheet etc. This operation is controller specific.
Table Block Count	number	Configures the table's block count (generally 10-15).
Learn Status	read-only	Reports the status of the network-level learn command (`busy`, `idle` or `error`).
Poll Frequency	drop-down list	Selects one of three pre-configured polling frequencies: `Slow`, `Normal`, `Fast`.

Property	Value	Description
Point Learn Count	read-only	Reports the number of points found during the latest discovery job.
Data Table Type	number	Defines the type of data table. This is a number.
Data Table Block	number	Defines the data table block number assigned to this Poc table (if it is a single block Poc table), otherwise this value is 255.
Data Table Start Block	number	Defines the starting data table block number assigned to this Poc table (if it is a multi-block Poc table), otherwise this value is 0.
Data Table End Block	number	Defines the data ending table block number assigned to this Poc table (if it is a multi-block Poc table), otherwise this value is 0.
Data Table Learn Count	number	Reports the number of data tables found during the latest discovery job.
Data Table Create Count	number	Reports the number of data tables created during the latest create process.

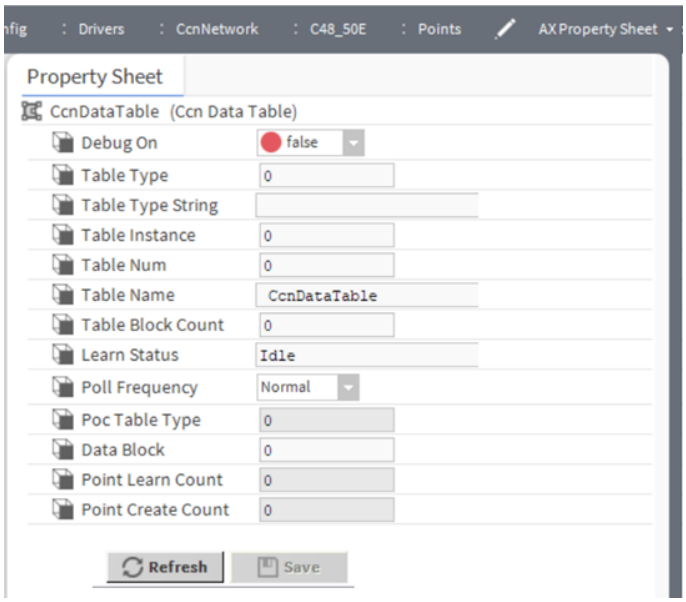
Actions

- **Fetch** retrieves additional table-related information.

ccn-CcnDataTable

This table configures a table with data-type proxy points.

Figure 5 Ccn Data Table properties



To access these properties, expand **Config→Drivers→CcnNetwork→CcnDevice→CcnPocTable**, double-click **CcnPocTable** and double-click a **CcnDataTable**.

Property	Value	Description
Debug On	`true` or `false` default	Turns debugging on (`true`) and off (`false`).
Table Type	number	Configures the type of table. For Data tables, valid type is >127.
Table Type String	text	Describes the table.
Table Instance	number	Assigns a table number to the table.
Table Num	number	Assigns a unique table number within **`Table Instance`**.
Table Name	text	Assigns an eight-character name to the table. This is the only property that goes to field device. To change the name in the device after entering it here, invoke the **Fetch** action on the table. An invalid table name is treated as "T." The driver sends the first eight characters of the name to the field device. **NOTE:** Change the **`Table Name`** property on the **Property Sheet** only. Avoid renaming the table name on the wire sheet, slot sheet etc. This operation is controller specific.
Table Block Count	number	Configures the table's block count (generally 10-15).
Learn Status	read-only	Reports the status of the network-level learn command (`busy`, `idle` or `error`).
Poll Frequency	drop-down list	Selects one of three pre-configured polling frequencies: `Slow`, `Normal`, `Fast`.
Poc Table Type	read-only	Displays the type of Poc table.
Data Block	number	Identifies the data block.
Point Create Count	number	Reports the number of points created during the latest create process.
Point Force Refresh	`true` or `false` default	Turns this function on (`true`) and off (`false`).

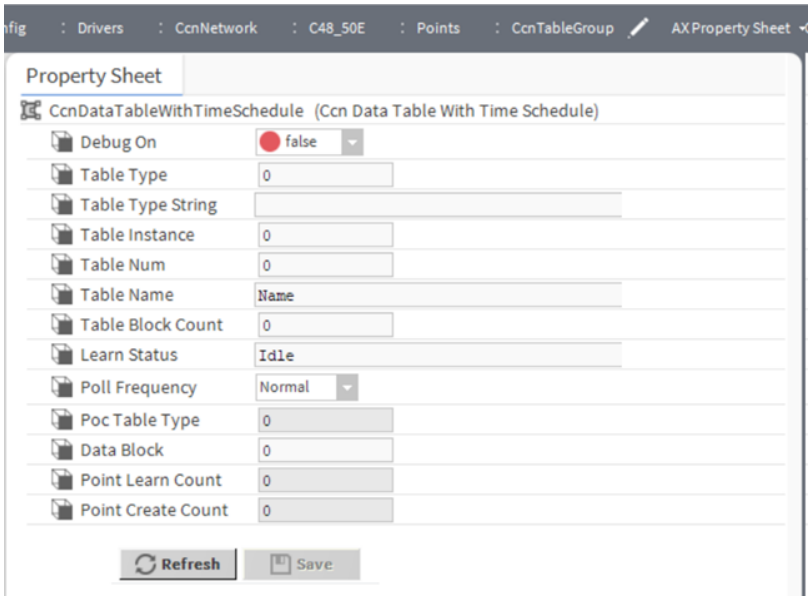
Actions

- **Fetch** retrieves additional table-related information.

ccn-CcnDataTableWithTimeSchedule

This component provides a data table with a time schedule.

Figure 6 Ccn Data Table with Time Schedule properties



To access these properties, expand **Config→Drivers→CcnNetwork→CcnDevice** and double-click **CcnPocTable**.

Property	Value	Description
Debug On	`true` or `false` default	Turns debugging on (`true`) and off (`false`).
Table Type	number	Configures the type of table. For Data tables, valid type is >127.
Table Type String	text	Describes the table.
Table Instance	number	Assigns a table number to the table.
Table Num	number	Assigns a unique table number within **`Table Instance`**.
Table Name	text	Assigns an eight-character name to the table. This is the only property that goes to field device. To change the name in the device after entering it here, invoke the **Fetch** action on the table. An invalid table name is treated as "T." The driver sends the first eight characters of the name to the field device. **NOTE:** Change the **`Table Name`** property on the **Property Sheet** only. Avoid renaming the table name on the wire sheet, slot sheet etc. This operation is controller specific.
Table Block Count	number	Configures the table's block count (generally 10-15).
Learn Status	read-only	Reports the status of the network-level learn command (`busy`, `idle` or `error`).
Poll Frequency	drop-down list	Selects one of three pre-configured polling frequencies: `Slow`, `Normal`, `Fast`.

Property	Value	Description
Poc Table Type	number	Displays the type of Poc table.
Data Block	number	Identifies the data block.
Point Learn Count	number	Reports the number of points found during the latest discovery job.
Point Create Count	number	Reports the number of points created during the latest create process.

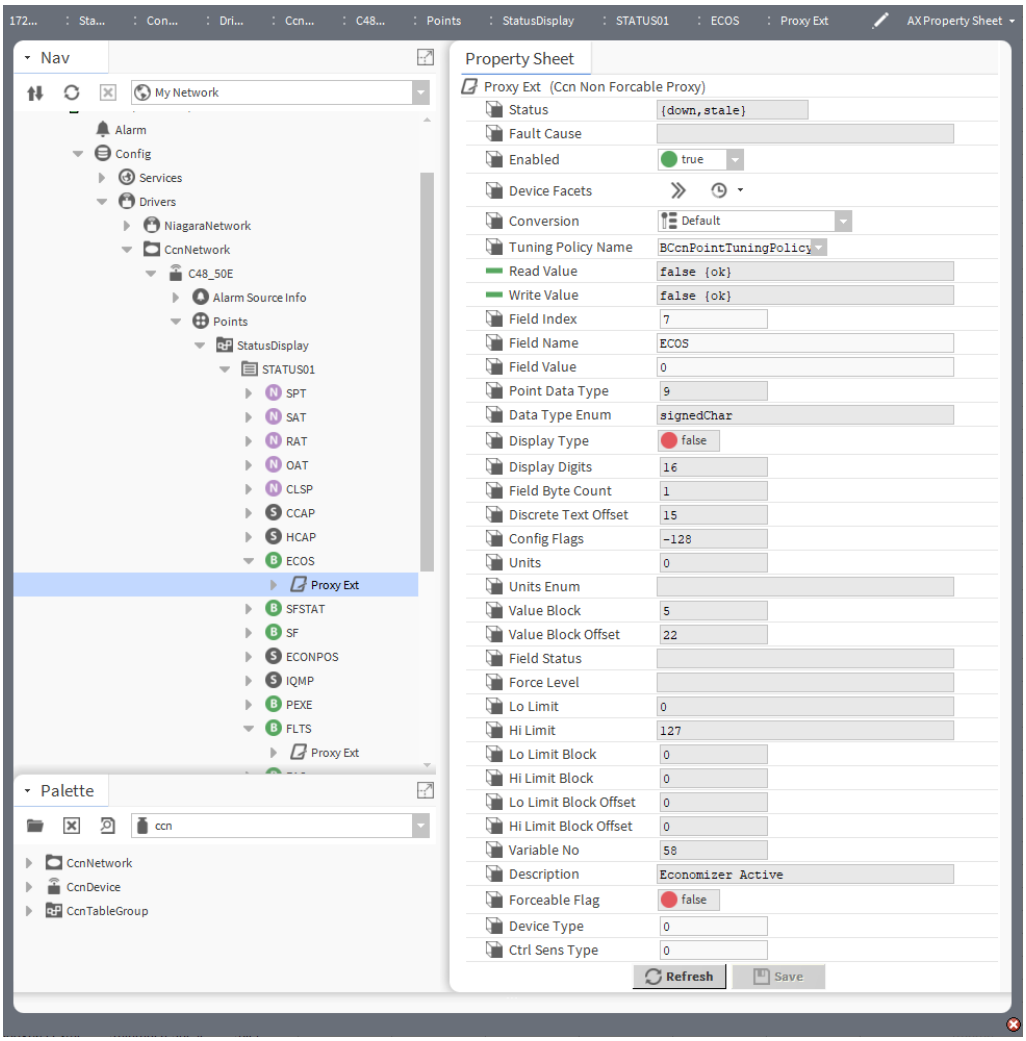
Actions

- **Fetch** retrieves additional table-related information.

Ccn-CcnObject

This component is a Boolean proxy point.

Figure 7 CcnObject or point's properties



To access these properties, expand **Config→Drivers→CcnNetwork→CcnDevice→CcnPocTable**, expand the Boolean point and double-click **Proxy Ext**.

Property	Value	Description
Field Name	text	Defines this point's name (8 ASCIIcharacters). Fid IO_Points Tables have up to 64 points. To modify each point name, change it from the **Property Sheet** and invoke the **Fetch** action.
Point Data Type	read-only	Configures the point's data type (00 – 33H).
Data Type Enum	read-only	Configures a point's data type enumeration. • 0, "eightBitFlags." • 1, "unsignedChar" • 2, "unsignedInt" • 6, "BEST_FloatingPoint" • 7, "IEEE_FloatingPoint" • 9, "signedChar" • 10, "signedInt" • 12, "timeInTwoBytes" • 16, "Name" • 17, "BCD" • 18,"controllerName" • 19,"controllerName" • 20,"controllerName" • 21,"controllerName" • 22,"pointName" • 23,"pointName" • 24,"pointName" • 25,"schedulePointNo" • 26,"schedulePointNo" • 27,"schedulePointNo" • 28,"schedulePointNo" • 29,"schedulePointNo" • 30,"schedulePointNo" • 31,"schedulePointNo" • 32, "phoneNumber" • 33, "password" • 34, "ASCII" • 48,"linkedFloatingPointValue" • 49,"numberOfDecimalPlaces"

Property	Value	Description
		• 50,"numberOfDecimalPlaces" • 51, "doubleTimeInFourBytes"
Display Type	read-only	Configures the type of display for each point from the device. `true` selects metric units. `false` selects imperial units. The database stores all device data values in imperial. The driver does not use this property, but instead uses the global property on the **CcnNetwork Config** tab named **`Display Metric`**.
Display Digits	read-only	Configures the point's display digit requirements. The upper nibble is the number of digits to the left of the decimal, and the lower nibble is the number of digits to the right of the decimal. For dataType 0 (eightBitFlags), the upper nibble specifies the number of usable bits in the byte (right to left). The driver displays this value in decimal and must be converted to hexadecimal to be interpreted (for instance a 97 decimal is a 61 hex). Thus, up to six digits display to the left of the decimal and one digit to the right.
Field Byte Count	read-only	Reports the point's byte count in the table value block.
Discrete Text Offset	read-only	Reports the point's discrete text offset if its discrete flag is set.
Config Flags	read-only	Reports the point's config flags. Valid values are: bit 0 – this point has a low limit bit 1 – this point has a high limit bit 7 – this point is a discrete point The driver displays this value in decimal, which must be converted to hexadecimal to be interpreted. For instance, a 128 decimal is an 80 hex and, thus, is a discrete point that does not have a low or high limit. Another common value, a 125 decimal is an 83 hex and, thus, is a discrete point that does have a low and high limit.
Units	read-only	Reports the point's value in integer units.
Units Enum	read-only	Reports the point's enumeration value in integer units.
Value Block	read-only	Reports the point's value block assignment.
Value Block Offset	read-only	Reports the point's value block offset assignment.
Lo Limit	read-only	Reports the point's lo limit value.
Lo Limit	read-only	Reports the point's lo limit block assignment.
Lo Limit Block Offset	read-only	Reports the point's lo limit block offset assignment.
Hi Limit	read-only	Reports the point's hi limit value.

Property	Value	Description
Hi Limit Block	read-only	Reports the point's hi limit block assignment.
Hi Limit Block Offset	read-only	Reports the point's hi limit block offset assignment.
Variable No	read-only	Reports the point's variable number (only applies to points under 11H, 12H, and 501H tables).
Description	read-only	Reports the point's 24-character description.
Forceable Flag	read-only	Reports the point's flags that can be forced. This value is `false` for a CcnInput and a CcnNonForceableOutput. This value is `true` for a CcnOutput if the point is in a 11H, 12H, or 501H table type, the point has a Hi Limit value and the point has a non-zero Variable No. CcnOutput's with a Forceable Flag. Force and Auto commands can set this value to `true`. This value is `false` for a CcnOutput (otherwise). The Set command can write a Ccn Output value with a Forceable Flag set to `false`.
Device Type	number	Assigns a numeric value to represent the device. **NOTE:** This property is not used. It is meant for the FID device.
Ctrl Sens Type	number	Assigns a numeric value. **NOTE:** This property is not used. It is meant for the FID device.

Chapter 5 Plugins

Topics covered in this chapter

- ♦ CCN Network views
- ♦ Ccn Device Manager
- ♦ Ccn Table Manager
- ♦ Ccn Point List Manager
- ♦ Ccn Data Points List Manager
- ♦ Ccn Alarm History Manager
- ♦ Ccn Fid Point List Manager
- ♦ Ccn Time Schedule Manager

There are many ways to view plugins (views). One way is directly in the tree. In addition, you can right-click on an item and select one of its views. Plugins provide views of components.

In Workbench, access the following summary descriptions on any plugin by selecting **Help→ On View** (F1) from the menu, or pressing F1 while the view is open.

CCN Network views

The **CcnNetwork** defaults to the **CCN Device Manager**, which is equivalent to the device manager in most other drivers. Use this view to discover and add **CcnDevice** components to a station.

Another view is the **Ccn Network View**, which is a **Property Sheet**.

Other standard views are also available on the **CcnNetwork**. However, apart from the **CCN Device Manager**, you typically access only its **Ccn Network View**.

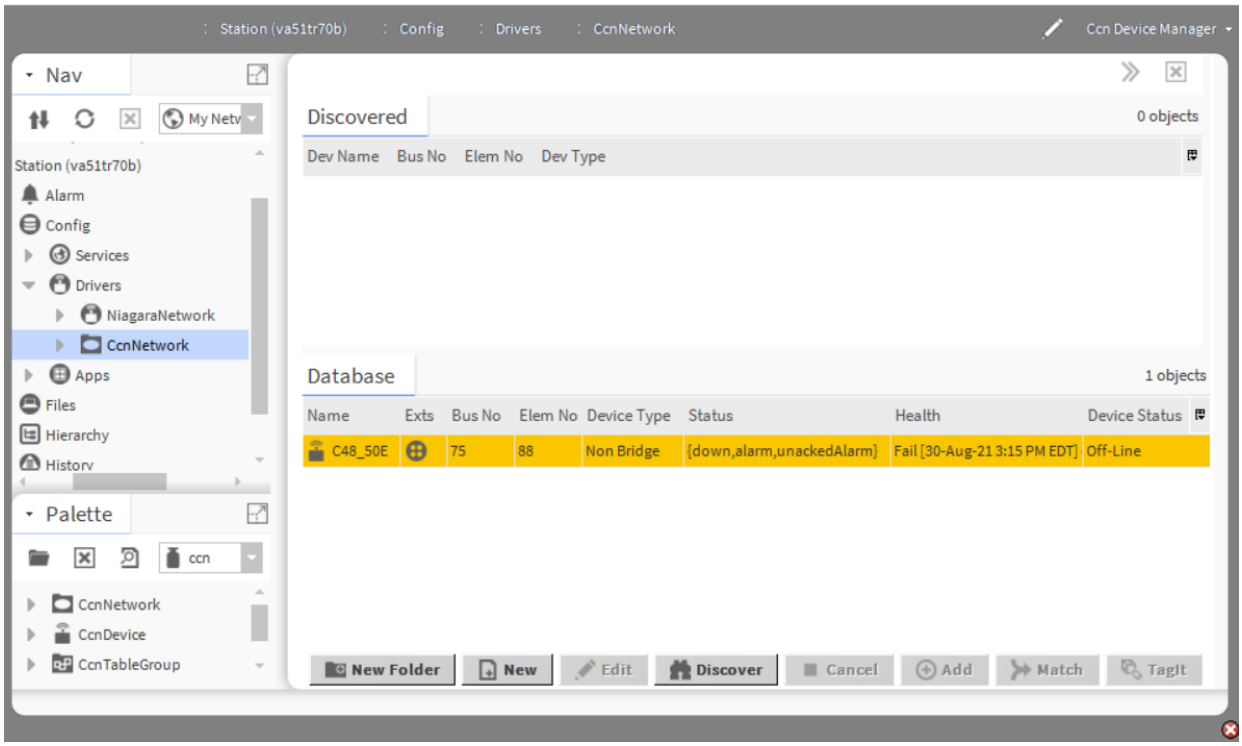
To open this view, expand **Config→Drivers** and double-click the **CcnNetwork**.

Ccn Device Manager

This manager provides a quick and easy way to display and learn CCN devices that are on the CCN network. Each row in this table-based view represents a unique device.

When building a network in the station, use this view to create, edit, and delete device-level components.

Figure 8 CCN Device Manager



To access the **CCN Device Manager**, expand **Config→Drivers** and double-click **CcnNetwork**.

The screen capture is an example of a **CCN Device Manager** for discovering and adding devices to the station database. It consists of either one or two main panes, depending on whether or not you clicked the **Discover** button.

Columns

Column	Description
Name	Reports the name of the Device.
Exts	Reports the point extensions.
Bus No	Reports the bus number
Elem No	Reports the element number.
Device Type	Reports the type of Device (`Unknown`, `Non Bridge` or `Bridge`)
Status	Indicates the current state of the Device.
Health	Reports the current health of the Device.
Device Status	Reports if the Device is Off-Line.

Buttons

- **New Folder** creates a new folder for devices. Each such folder provides its own set of manager views.
- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.

- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** button is not used by the CCN driver.
- **TagIt** associates metadata, such as location or unique configuration with the object.

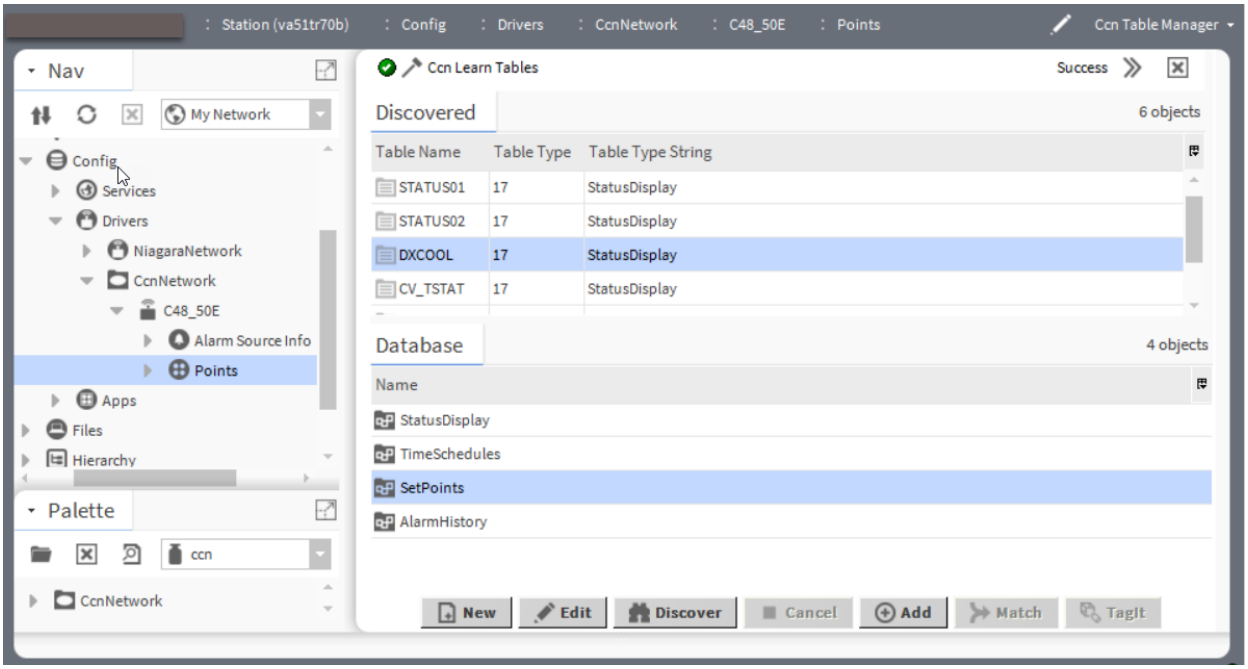
Learn Status property in the **Ccn Network** view indicates the progress of the discovery job. Once the discovery job is complete, the **Discovered** pane appears in the view pane of the **CCN Device Manager** displaying a table of discovered devices.

Ccn Table Manager

This view provides a quick and easy way to display and learn CCN tables that are on the **CcnDevice**.

Each row in this view represents a unique table. When you build a device in the station, use this view to create, edit, and delete table-level components.

Figure 9 Ccn Table Manager with Discovered pane



To open the **Ccn Table Manager**, expand **Config→Drivers→CcnNetwork→CcnDevice** and double-click the **Points** folder. It is also the view that opens when you double-click a **CcnTableGroup** in the Nav tree.

Columns

Column	Description
Name	Reports the name of the entity or logical grouping.
Table Name	Reports the name of the table.
Table Type	Reports the type of table.
Table Type String	Reports more information about the table type.

Buttons

- **New** creates a new device record in the database.

- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** button is not used by the CCN driver.
- **TagIt** associates metadata, such as location or unique configuration with the object.

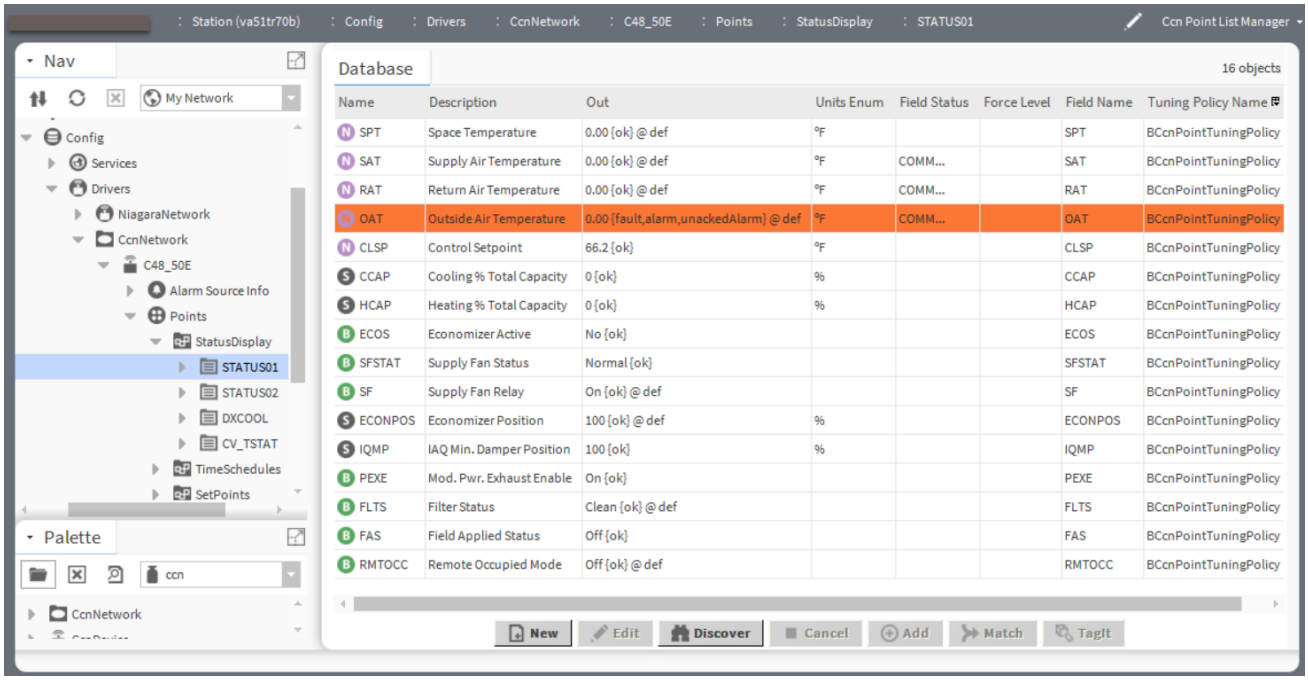
Ccn Point List Manager

This is the default view of a **Ccn Point List Manager**. This manager view provides a quick and easy way to display and learn CCN points that are in the CCN PIC table.

To open the **Ccn Point List Manager**, expand **Config→Drivers→CcnNetwork→CcnDevice→Points→Ccn-TableGroup** and double-click **CcnPicTable**. The **Learn Status** property in the **Ccn Network** view indicates the progress of the discovery job. Once the discovery job is complete, the top half-pane of the point manager displays a table of discovered devices.

The **Learn Status** property in the **Ccn Network** view indicates the progress of the discovery job. Once the discovery job is complete, the top half-pane of the **Ccn Point List Manager** displays a table of discovered devices.

Figure 10 CCN Point List Manager



Columns

Column	Description
Name	Reports the name of the Point List Manager.
Description	Reports the description of the Point List Manager.
Out	Reports the point value when the discovery job executed.

Column	Description
Units Enum	Reports the point's units enum value.
Field Status	Reports the points fields status value.
Force Level	Reports the points force level value.
Field Name	Reports the field name.
Tuning Policy Name	Reports the name of the Tuning Policy or the default policy.

Buttons

- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** button is not used by the CCN driver.
- **TagIt** associates metadata, such as location or unique configuration with the object.

Ccn Data Points List Manager

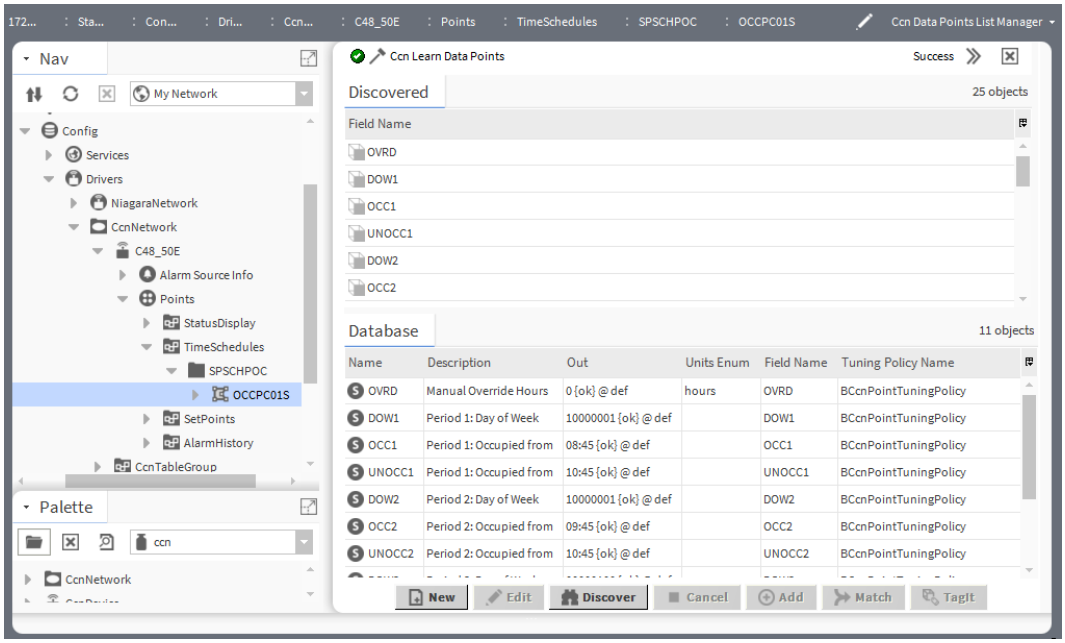
This view provides a quick and easy way to display and learn CCN points that are on the CCN POC table.

To open the **Ccn Data Points List Manager**, expand **Config→Drivers→CcnNetwork→CcnDevice→-Points→CcnTableGroup** and double-click the **CcnDataTable** in the Nav tree.

Each row in this table-based view represents a unique point. When you build a device in the station, use this view to create, edit, and delete point-level components. The **Learn Status** property in the **Ccn Network** view indicates the progress of the discovery job. Once the discovery job is complete, the top half-pane of the point manager displays a table of discovered devices.

The **Learn Status** property in the **Ccn Network** view indicates the progress of the discovery job. Once the discovery job is complete, the top half-pane of the **Ccn Data Points List Manager** displays a table of discovered devices.

Figure 11 Ccn Data Points List Manager



Columns

Column	Description
Name	Reports the name of the Point List Manager.
Description	Reports the description of the Point List Manager.
Out	Reports the point value when the discovery job executed.
Units Enum	Reports the points units enum value.
Field Name	Reports the points field name.
Tuning Policy Name	Reports the name of the Tuning Policy or the default policy.

Buttons

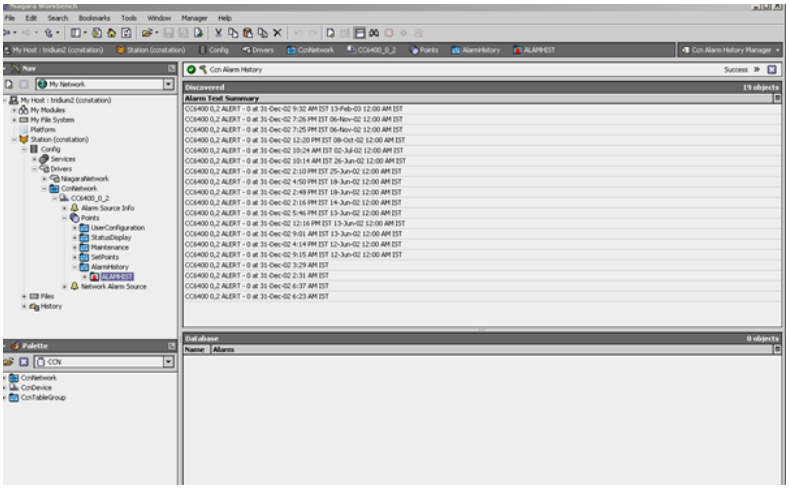
- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** button is not used by the CCN driver.
- **TagIt** associates metadata, such as location or unique configuration with the object.

Ccn Alarm History Manager

This is the default view of the **Ccn Alarm History Manager**.

To open the **Ccn Alarm History Manager**, expand **Config→Drivers→CcnNetwork→CcnDevice→-Points→CcnTableGroup** and double-click the **CcnAHTable** in the Nav tree.

Figure 12 Ccn Alarm History Table



Columns

Column	Description
Name	Reports the name of the Alarm
Alarm	Reports the Alarm class

Buttons

- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** button is not used by the CCN driver.
- **TagIt** associates metadata, such as location or unique configuration with the object.

Ccn Fid Point List Manager

This is the default view of the **Ccn Fid Point List Manager**. It provides a quick and easy way to display and learn CCN points that are in a CCN FID table.

Each row in this table-based view represents a unique point. When you build a device in the station, use this view to create, edit, and delete point-level components.

The **Learn Status** property in the **Ccn Network** view indicates the progress of the discovery job. Once the discovery job is complete, the top half-pane of the **Ccn Fid Point List Manager** displays a table of discovered devices.

Figure 13 Point's discovery on a FID table

Columns

Column	Description
Name	Reports the name of the Fid Point List Manager.
Description	Reports the description of the Fid Point List Manager.
Out	Reports the current out value, including any point facets. This defaults to the single (configured) property value along with status for the proxy point.
Units Enum	Reports the points units enum value.
Field Name	Reports the field name of Fid Point List Manager.

Buttons

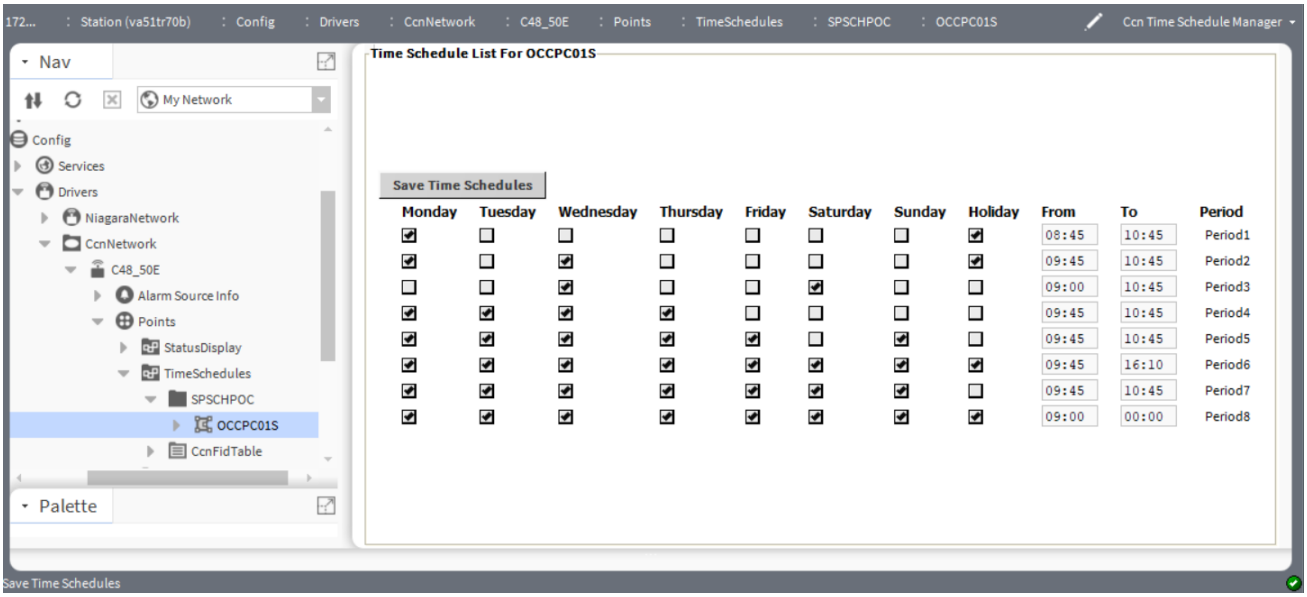
- **New** creates a new device record in the database.
- **Edit** opens the device's database record for updating.
- **Discover** runs a discover job to locate installed devices, which appear in the **Discovered** pane. This view has a standard appearance that is similar to all **Device Manager** views.
- **Cancel** ends the current discovery job.
- **Add** inserts into the database a record for the discovered and selected object.
- **Match** button is not used by the CCN driver.
- **TagIt** associates metadata, such as location or unique configuration with the object.

Ccn Time Schedule Manager

This is a tabular view of the table types DataTableWithTimeSchedule and FIDTableWithTimeSchedule. This view allows you to select the week days and change the from-to time.

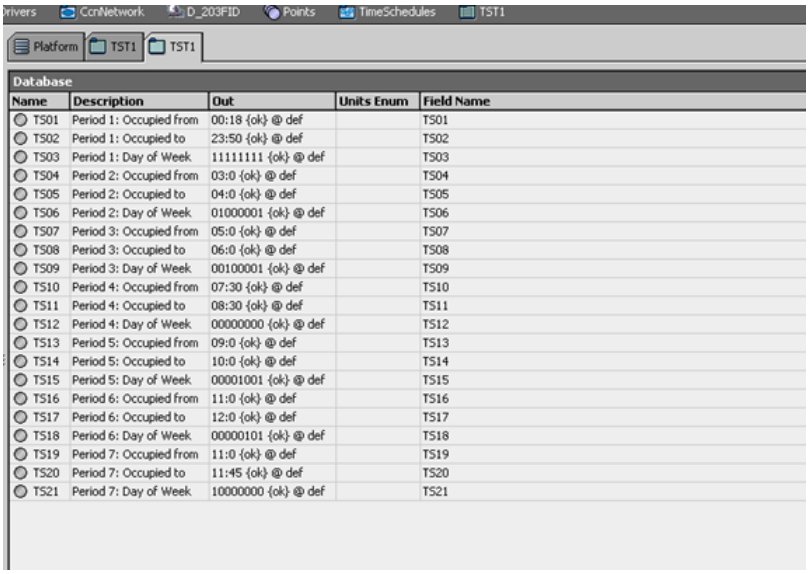
To open the **Ccn Time Schedule Manager**, expand **Config→Divers→CcnNetwork→CcnDevice→Points→TimeSchedules→CcnDataTable**. Right-click **CcnDataTableWithTimeSchedule** and select **Views→Ccn Time Schedule Manager**.

Figure 14 CCN Time Schedule Manager



This screen capture is an example of a **CCN Time Schedule Manager**.

Figure 15 CCN Fid Point List Manager



This screen capture is an example of a **Ccn Fid Point List Manager** view.

The **Save Time Schedules** button saves the selected values to the station database and reflects the changes in the **Ccn Fid Point List Manager** view.

The **Ccn Time Schedule Manager** reflects the changes made in the **Ccn Fid Point List Manager** and vice versa.

Index