Andover Controls Infinet Driver for Tridium Niagara

User Guide

baudrate.io

niagara4

February 22, 2026

Contents

1 Introduction **2**

2 Requirements **4**

3 Primary Network Controller **4**

4 Quick Start **5**

5 Infinet Networks **6**

6 Infinet Devices **7**

7 Infinet Points **8**

8 Tuning Policies **10**

9 Infinet Histories **11**

10 FAQ **12**

10.1 I cannot establish communication. What should I check? 12

10.2 I can discover devices, but their status is Offline. 12

10.3 Does the driver support Xdriver devices? 12

10.4 Differences between Infinet driver and Infinity driver? 13

1 Introduction

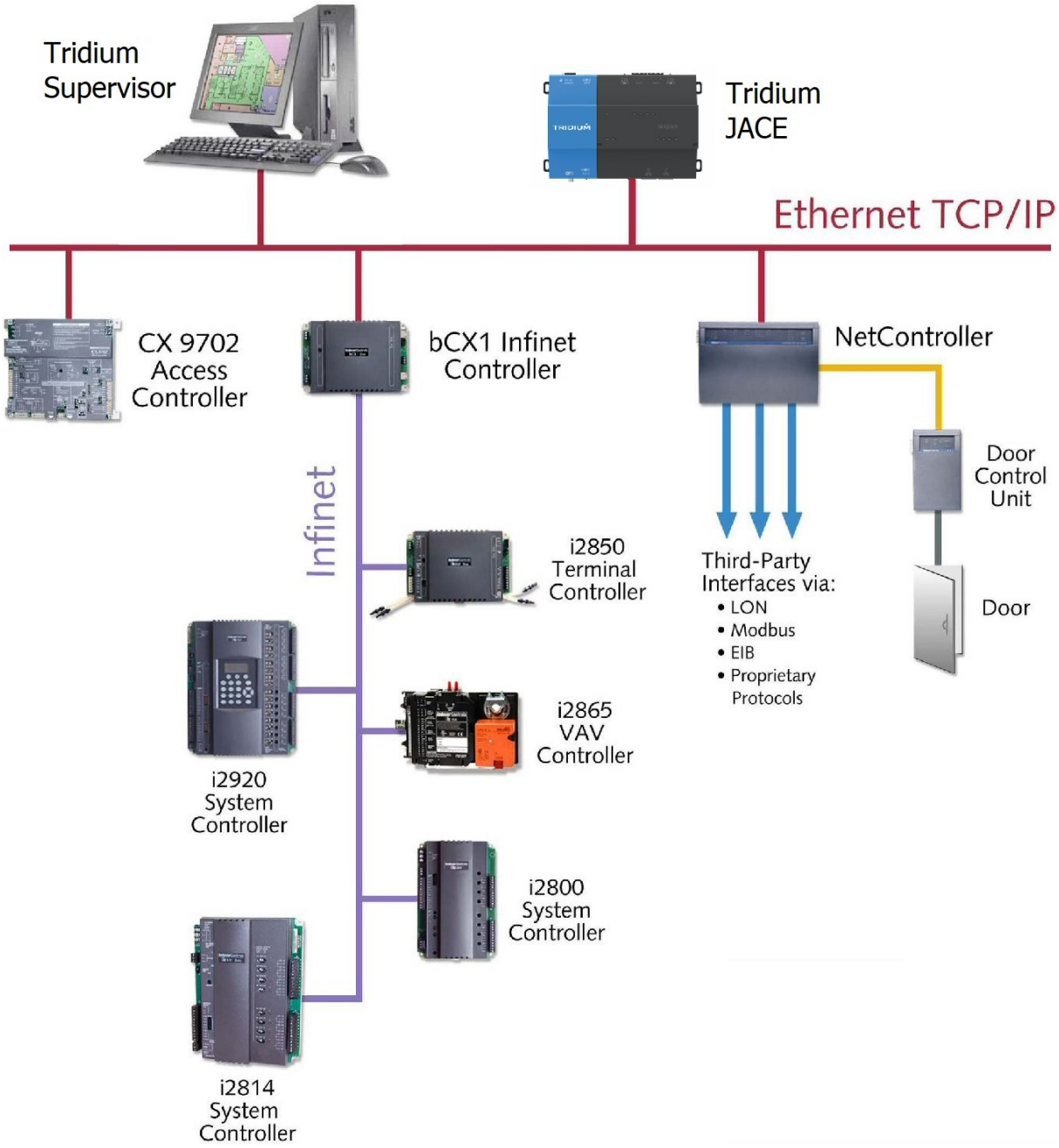

Figure 1: Infinet architecture

Infinet is a communication protocol for Andover Controls[1] (now part of Schneider Electric) Infinity programmable controllers. Infinet controller family includes:

- Ethernet network controllers, such as bCX1 and NetController 2 (CX9680). These devices act as global strategy controllers, protocol gateways, email servers, and field controller interfaces with up to 127 field controllers per RS-485 port. They can also be equipped with add-on IO modules. Network controller models:

ACX5720	ACX5721	ACX5740	ACX5741	CX9200	CX9300
CX9400	CX9401	CX9410	CX9411	CX9680	CX9681
CX9702	CX9900	CX9901	CX9924	CX9940	CX9960
b4920	bCX4000	bCX4020	bCX4040	bCX9600	bCX9640

- RS-485 field controllers, such as i2800. There are freely programmable controllers with various numbers of IO points. There are original Infinity and newer Infinity 2 field controllers, including series:
  - Display unit (DCX)
  - Access controllers (ACX)
  - Local controllers (LCX, i26xx, i280x, i281x)
  - Terminal controllers (TCX, i285x, i286x, i288x)
  - Lighting controllers (LCX89x)
  - System controllers (SCX, i2920)

DCX250	ACX700	ACX701	ACX780	ACX781	LCX800
LCX800i	LCX810	TCX840	TCX841	TCX842	TCX843
TCX844	TCX845	TCX846	TCX847	TCX848	TCX849
TCX850	TCX851	TCX852	TCX853	TCX855	TCX860
TCX861	TCX862	TCX863	TCX864	TCX865	TCX866
TCX867	TCX868	TCX869	TCX870	LCX890	LCX891
LCX892	LCX898	SCX900	SCX920	i2608	i2616
i2624	i2800	i2804	i2810	i2814	i2850
i2851	i2853	i2865	i2866	i2867	i2885

[1] All trademarks or registered trademarks are the property of their respective owners

i2886	i2887	i2888	i2889	i2920

Infinet driver for Tridium Niagara uses the same communication protocol as Andover BMS software Continuum Cyberstation. The driver connects to network controllers via TCP/IP interface, so no extra cabling is required. Controller software should not be modified, all devices, points, and histories are discovered automatically. Niagara points are displayed with names, descriptions, units, and precision, which greatly simplifies the integration process.

In addition to building control, Andover Infinity family also includes access control devices. Note: the current revision of Infinet driver does not support access-specific objects, like door, personnel, area. If your project requires them, e-mail us to discuss.

For more information about Andover Continuum hardware and software refer to Schneider Electric website.

2 Requirements

- Niagara-powered device with software v4.8 or later, such as JACE, Supervisor, or any OEM version
- Infinet driver module and license
- IP address and network address of the primary network controller

3 Primary Network Controller

To discover all network and field devices automatically, the driver requires that you designate one Infinity network controller as the *Primary Controller*. The driver gets information about the network through this controller during device discovery. The primary controller must have had the network *taught* to it before the integration. This is done through the *Teach* function in the Continuum Cyberstation.

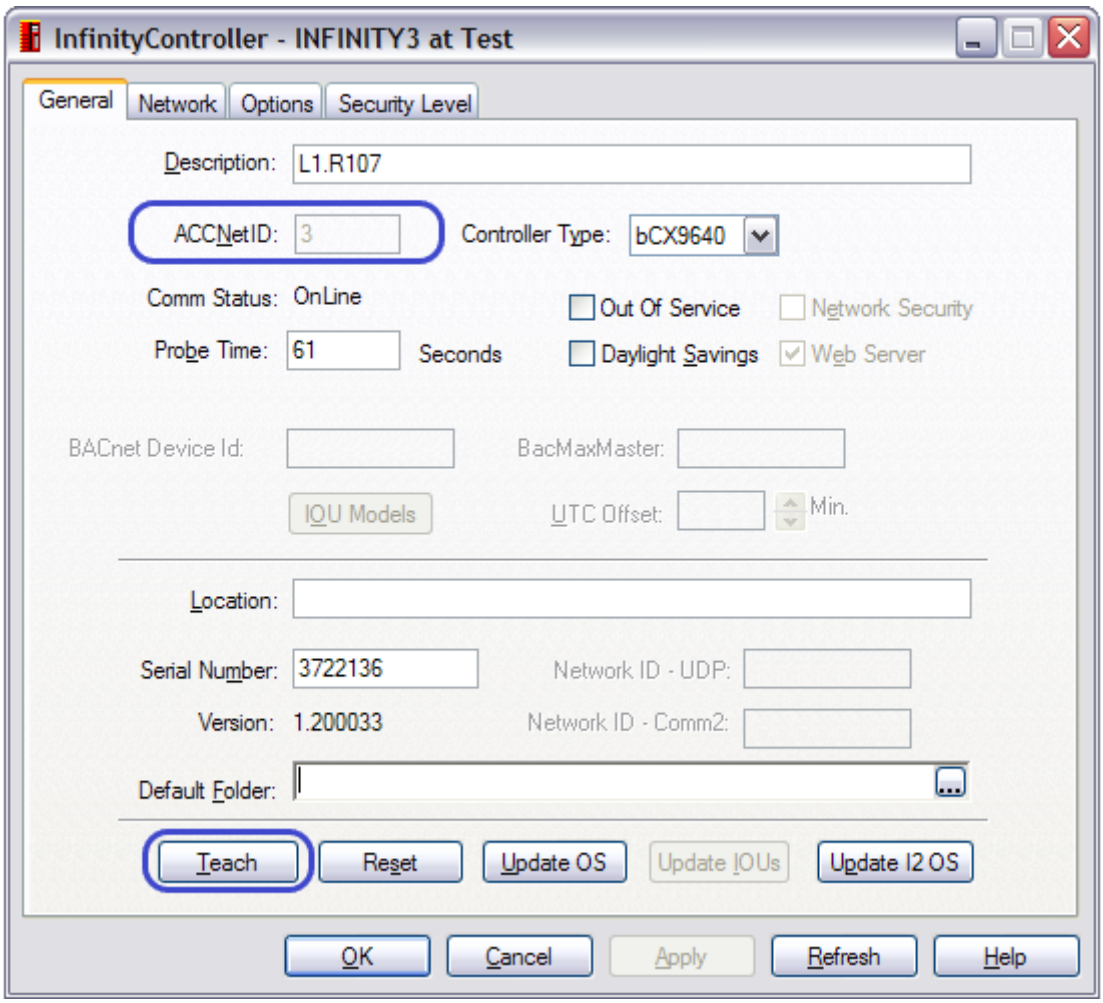

Figure 2: Teaching network controller

In addition to IP address (displayed on the Network tab), it is necessary to determine *network id* number, sometimes called *ACCNet ID*. It can be found in Cyberstation or on the controller web page.

4 Quick Start

1. Copy infinet-rt.jar module to Niagara /modules folder of your PC and restart the software.
    a. If the driver is hosted on PC: restart Supervisor station.
    b. If the driver is hosted on JACE: import the module with all its dependencies into JACE and restart it.
2. Allow incoming UDP packets in PC firewall, if the driver is hosted on PC or on JACE with Niagara

4.15 or later (use **FirewallPortHelper** from **net** palette).

3. Add **Infinet Network** to the station. Note: do not confuse it with Infinity Network.
4. Enter license code in **License** property under **Infinet Network**.
5. Enter primary controller **Ip Address** and **Network Id** under **Infinet Network / Comm Config / Primary Address**.
6. **Discover** devices – you should be able to see available network controllers and, under them, field controllers.
7. Add discovered devices to the station.
8. Open device **Points** extension and **Discover** points.
9. Add points to the station. By default, all points are added as read-only – change type to writable if required.

5 Infinet Networks

Infinet Network contains many standard Niagara properties, as well as a few driver-specific ones:

- **License** – the code that allows the driver to run on your host.
- **Comm Config / Primary Address** – IP address and network id of the primary IP controller.
- **Comm Config / Response Timeout** – the number of milliseconds Niagara waits for controller response. Default 500ms value can be increased in case of timeout errors.
- **Comm Config / Retry Count** – how many times Niagara repeats requests if there is no response from the controller.
- **Poll Scheduler** – specifies polling rates and displays network communication metrics. This information is helpful to see if the network is "healthy".
- **Tuning Policies** – see Tuning Policies section.

In most cases only **License** and **Comm Config / Primary Address** values need to be changed.

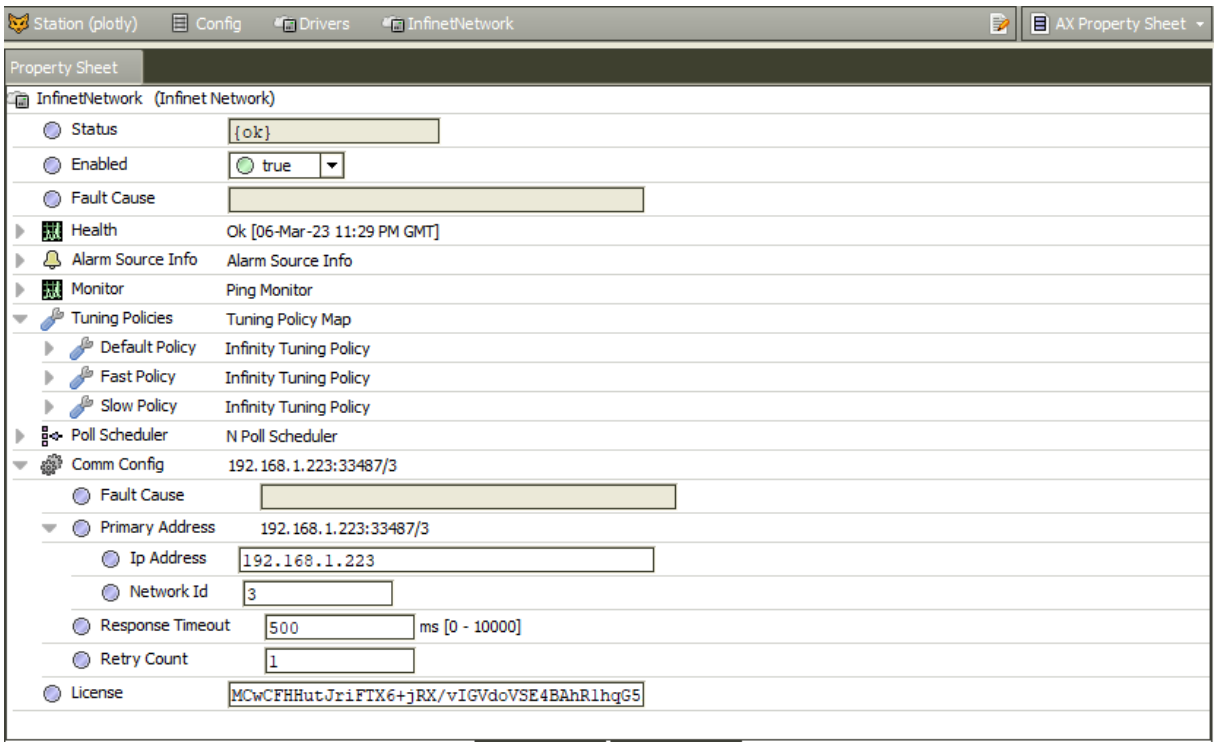

Figure 3: Infinet network properties

After Infinet network properties are set, **Infinet Device Manager** view is used to start **Discover** devices. The discovery process can be tracked with a progress bar and the found devices can be seen by pressing **»** button.

6 Infinet Devices

After the device discovery, the top pane displays all found network controllers. Field controllers are located underneath and can be seen by pressing **+** symbol. Each found controller is displayed with the name, IP address, network id, device model, serial number, and description. If the controller is known to the primary network controller, but the driver cannot establish communication with it, then the description field will show *Offline*.

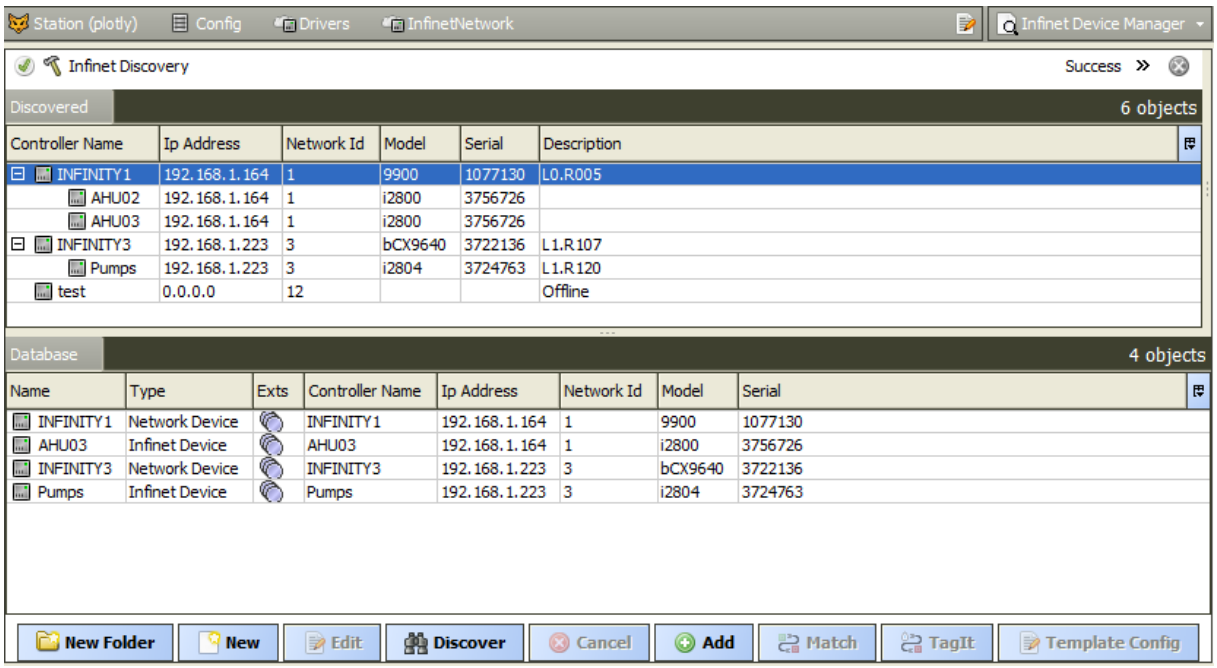

Figure 4: Infinet devices

Devices are added to Niagara by dragging from the top pane or by selecting and pressing **Add** button. There are two types of devices: **Network Device** (IP controller) and **Infinet Device** (field controller). Properties **Controller Name**, **Ip Address**, and **Network Id** identify the device on Infinet network and allow Niagara to communicate with it. Although it is possible to create a device manually with **New** button, it is recommended to use device discovery to ensure these properties are set correctly. Property **Name** can be any string and properties **Model** and **Serial** are read from the device for informational purposes.

7 Infinet Points

Infinet points are discovered automatically in **Infinet Point Manager** view. Each discovered point has the following properties:

- **Point Name** – how the point is called in Infinity controller.
- **Point Type** – Input, Output, Numeric (software variable), String, or System Variable. System Variables are built-in points for device information like date, time, and controller memory size.
- **Elec Type** – IO configuration.
- **Channel** – IO terminal number.
- **Value** – real-time point value.

- **Units** – physical units.
- **Format** – display format.
- **Description** – point description.

Properties **Units**, **Format**, and **Description** are only displayed if they are configured in the controller.

Points are added to Niagara by dragging from the top pane or by selecting and pressing **Add** button. Each added point contains the following properties:

- **Name** – any string to identify the point in Niagara.
- **Point Name** – string to address the point in the controller. Must not be changed.
- **Type** – Niagara point type: Boolean, Numeric, Enum, or String Point. The recommended point type is automatically determined by the driver, but it can be changed during point creation. Infinity software variables can also be created as writable points, so they can be changed from Niagara.
- **Tuning Policy Name** – allows assigning a polling rate from the tuning policy.

Driver automatically assigns point facets – units and precision – based on **Point Type**, **Elec Type**, **Units**, and **Format**.

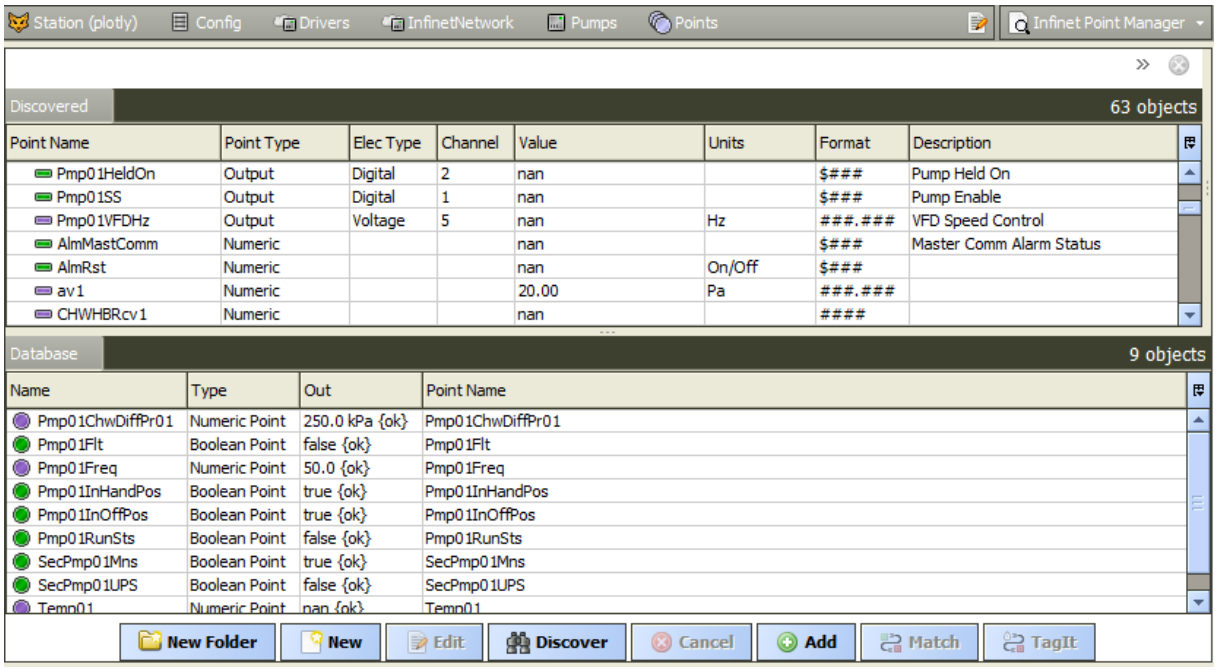

Figure 5: Infinet points

8 Tuning Policies

Tuning Policies define various aspects of point reading and writing. Each policy contains the following properties:

- **Min Write Time** – applies to writable proxy points, especially ones that have one or more linked inputs. Specifies the minimum amount of time allowed between writes. Provides a method to throttle rapidly changing value so that only the last value is written. If this property value is 0 (default), this rule is disabled (all value changes attempt to write).
- **Max Write Time** – applies to writable proxy points. Specifies the maximum "wait time" before rewriting the value, in case nothing else has triggered a write. Any write action resets this timer. If property value is 0 (default), this rule is disabled (no timed rewrites). Exercise caution to avoid "write contention" that can result in toggling loads.
- **Write On Start** – applies to writable proxy points. Determines behavior at station startup. If true, a write occurs when the station first reaches "steady state."
- **Write On Up** – applies to writable proxy points. Determines behavior when proxy point (and parent device) transitions from "down" to "up". If true, a write occurs when the parent device transitions from down to up.
- **Write On Enabled** – applies to writable proxy points. Determines behavior when a proxy point's status transitions from "disabled" to normal (enabled). If true, a write occurs when the writable point transitions from disabled.
- **Stale Time** – if set to a non-zero value, points become "stale" (status stale) if the configured time elapses without a successful read, indicated by Read Status "ok". If set to zero (default), the stale timer is disabled.
- **Read Override** – if true, Niagara reads Andover *state* property to determine whether this point is controlled by a program or is overridden. Andover *state* is mapped into Niagara *overridden* status. It allows users to see if the point is overridden from Continuum Cyberstation front-end. The downside of enabling this property is that it produces two polling requests (value and state) instead of one and increases the traffic.
- **Write Override** – if true, Niagara changes Andover *state* property according to point *overridden* status. It allows overriding every point in Andover controllers: inputs, outputs, and software variables. Can be used for IO testing.
- **Poll Frequency** – provides a method to associate the tuning policy with one of 3 Poll Rates available in the network's **Poll Scheduler**: Fast Rate, Normal Rate, or Slow Rate.

Initially there are four policies created, but more can be added:

- **Default Policy** – 10 seconds.
- **Fast Policy** – 5 seconds.

- **Slow Policy** – 30 seconds.
- **Override Normal Policy** – 10 seconds, read and write override enabled.

9 Infinet Histories

Infinet controllers can collect logs locally, so the client software – Continuum Cyberstation or Niagara Infinet driver – can periodically import them as histories for viewing and analysis. It is recommended to import histories from controllers instead of collecting them at Niagara station with a history extension: this way historical data is retained even if the Niagara station restarts, and Niagara does not have to subscribe to all logged points, allowing more bandwidth for real-time values.

The Infinet logs are discovered automatically in **Infinet History Manager** view. Each discovered log has the following properties:

- **Point Name** – how the point with history is called in Infinity controller.
- **Point Type** – Input, Output, Numeric (software variable), String, or System Variable.
- **Log Type** – Manual, Instantaneous, Average, Minimum, or Maximum. In manual logs samples are written by Continuum programs. In average, minimum, and maximum logs, each sample is calculated during the last interval.
- **Size** – the number of samples stored in each log.
- **Interval** – the time interval between each log sample.
- **Last Timestamp** – the timestamp of the last stored sample.

Logs are added to Niagara by dragging from the top pane or by selecting and pressing **Add** button. Each log is mapped into **Infinet History Import** object that contains the following properties:

- **Name** – any string to identify the history import in Niagara.
- **History Id** – names of the history folder and the history.
- **Execution Time** – how often Niagara imports the history from the controller. The interval should be selected taking into consideration log size and sampling interval to avoid gaps in the history.
- **Enabled** – allows disabling history import.
- **Capacity** – whether the number of samples stored in Niagara should be limited.
- **Full Policy** – in case of limited capacity, what should be done with old data.
- **Point Name** – string to address the point in the controller. Must not be changed.

After **Infinet History Import** is created, it can be executed – automatically, according to **Execution Time**, or manually with **Execute** action or **Archive** button – to create Niagara history and fill it with all values stored in the controller. During further imports, only the new samples will be collected.

10 FAQ

10.1 I cannot establish communication. What should I check?

- Try to ping the primary network controller IP from Niagara network.
- Check if the license is valid and has not expired. Restart the station if expired.
- In case the driver is deployed on Supervisor, check if the computer firewall allows UDP traffic.
- If the driver is deployed on JACE with Niagara 4.15 or later, add **FirewallPortHelper** component from **net** palette to the station and set it to allow UDP port 33487.
- Check if the primary controller IP address and network id in Niagara are the same as in the primary controller.
- Try another network controller as a primary controller.
- By default, Infinity network controllers communicate via UDP protocol, but it can be changed in Cyberstation controller Network tab. If TCP or TCP/UDP is selected, change it back to UDP.
- Enable Infinet logs (right-click on Station => Spy => logSetup, change all Infinet.* logs to ALL), try to discover devices and send the contents of Application Director to support@baudrate.io.

10.2 I can discover devices, but their status is Offline.

That might happen on Supervisor or on JACE starting from Niagara 4.15 when its firewall blocks all incoming UDP packets. The driver sends requests to UDP port 33487, but some Andover controllers reply from randomly selected ports. Firewall software, e.g., Windows Defender, normally blocks all such messages, so Niagara does not receive any replies and assumes the controllers are offline.

Allow incoming UDP packets on all ports of Supervisor machine.

10.3 Does the driver support Xdriver devices?

Andover Xdrivers are software modules that can be deployed in Infinity network controllers to support third-party protocols, such as Modbus. A network controller with an Xdriver can read and write values from non-Infinity devices via IP or serial interface.

Infinet driver for Niagara communicates with Infinity network controllers and discovers all points, including the ones created for Xdriver. Niagara can see all Xdriver points and Xdriver functionality is fully supported.

10.4 Differences between Infinet driver and Infinity driver?

Tridium Infinity driver allows Niagara to connect to Andover NetController via serial RS-232 port and to read points from its service console. This driver does not work with newer Infinity devices, it requires service port configuration, can only access field devices connected to one NetController and its performance is low due to limited RS-232 speed.

Baudrate Infinet driver communicates via Ethernet, uses the same protocol as native Andover software, and can access all network and field controllers. Infinity controllers do not need to be reconfigured, and no extra cabling is required. Niagara can be installed on the same LAN or remotely. Driver performance is high due to Ethernet speed.