Siemens Apogee P1 Driver for Tridium Niagara

User Guide

baudrate.io

niagara4

September 11, 2026

Contents

1 Introduction **2**

2 Requirements **2**

3 Quick Start **2**

4 Apogee P1 Network **3**

5 Apogee P1 Devices **5**

6 Apogee P1 Points **6**

6.1 Point Types 7
6.2 Point Facets 8
6.3 Value Scaling 8
6.4 Proxy Extension Properties 9

7 Writing and Releasing Points **9**

7.1 Writing Values 9
7.2 Releasing Overrides 9

8 FAQ **10**

8.1 A device appears twice in discovery 10
8.2 I cannot discover any devices 10
8.3 Points are not reading 10
8.4 A value I wrote is lost after the controller is power cycled 10
8.5 Points intermittently go to fault on a busy trunk 11
8.6 Communication errors or CRC errors 11

1 Introduction

The Apogee P1 driver provides communication with Siemens Apogee[1] FLN (Field-Level Network) devices using the P1 protocol. The P1 protocol is a master-slave, half-duplex protocol running over RS-485. It is used by Siemens building automation controllers including TEC (Terminal Equipment Controller) and other Apogee field devices.

The driver supports automatic device and point discovery, real-time point polling, writing values, and releasing overrides. Discovered points are displayed with names, types, engineering units, and boolean state labels, which greatly simplifies the integration process.

Supported devices include:

- Siemens Apogee TEC controllers (all revisions)
- Siemens LAB (Laboratory Controller)
- Other FLN/P1 protocol compatible devices

Parameter	Value
Physical Layer	RS-485 (2-wire shielded twisted pair)
Baud Rate	4800 bps (default, configurable)
Data Format	8 data bits, No parity, 1 stop bit
Device Address Range	1-99
Max Devices per Trunk	32 (recommended)

2 Requirements

- Niagara-powered device with software v4.8 or later, such as JACE, Supervisor, or any OEM version
- Apogee P1 driver module and license
- RS-485 serial port
- P1/FLN network with devices configured with unique addresses

3 Quick Start

1. Copy apogeeP1-rt.jar module to Niagara /modules folder and restart the software.

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

   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. Add **Apogee P1 Network** to the station from the **apogeeP1** palette.
3. Enter license code in **License** property under **Apogee P1 Network**.
4. Configure serial port in **Serial Config** property:
   - Set **Port Name** to the COM port (e.g., "COM1").
   - **Baud Rate** defaults to 4800. Some controllers may use a different baud rate – adjust if needed.
5. **Discover** devices – right-click network and select Discover.
6. Add discovered devices to the station.
7. For each device, open **Points** extension and **Discover** points.
8. Add points to the station. The driver automatically selects the correct Niagara point type (read-only or writable) based on the P1 point type.

4 Apogee P1 Network

Apogee P1 Network contains standard Niagara network properties plus driver-specific ones:

- **License** – the code that allows the driver to run on your host. Without a valid license, the network will show a configuration fault.
- **Serial Config** – serial port configuration with the following settings:
  - **Port Name** – COM port name (e.g., "COM1").
  - **Baud Rate** – communication speed. Default is 4800 bps, which is the standard for most P1 devices. Some controllers may require a different baud rate.
  - **Receive Timeout** – time in milliseconds to wait for a device response (default 250ms). Can be increased for slow networks.
  - **Retry Count** – number of retries on communication failure (default 1).
- **Tuning Policies** – defines polling and write behavior for points. Each policy includes standard Niagara tuning properties (stale time, write timing, etc.) plus three driver-specific ones:
  - **Poll Frequency** – Fast, Normal, or Slow, corresponding to the **Poll Scheduler** rates.
  - **Read After Write** – when enabled, the driver reads the point value immediately after a write to confirm the device accepted the change. Every write then costs two exchanges on the trunk instead of one. Turn it off for points written by logic: confirming a value that is about

to change again achieves little, and the next poll reports what the device actually holds. Keep it on for points written by hand, where the confirmation is worth the extra traffic.

- **Write To EEPROM** – controls whether a written value survives a power cycle. When disabled (the default), the value takes effect immediately but is lost if the controller restarts. When enabled, the device also stores it permanently. That is the only difference – in both cases the write takes effect at once.

  Device storage accepts a limited number of writes before it wears out, so enable this only for points that are written occasionally, such as a setpoint someone types in. Never enable it on a point linked to a control loop, a schedule, or any other logic: such a point changes continuously, and storing every change will wear the device out within days and damage it permanently. A few points refuse to be written without being stored and go to fault; those need this enabled regardless, so keep their write rate low.

Four policies are provided by default:

- **Normal Policy** – poll frequency Normal (30 seconds), read after write enabled.
- **Fast Policy** – poll frequency Fast (10 seconds), read after write enabled.
- **Slow Policy** – poll frequency Slow (1 minute), read after write enabled.
- **Persistent Write Policy** – poll frequency Normal (30 seconds), read after write enabled, and **Write To EEPROM** enabled so written values survive a power cycle. Use it only for points that need that, such as setpoints entered by an operator, and never for points written by automatic logic.

All four ship with **Min Write Time** set to 5 seconds, so a point driven by automatic logic sends at most one write per 5 seconds, carrying the most recent value. Without it a linked control source can write several times a second: it writes on every change of its output, and small changes that round to the same raw value still produce a write. Raise it for points that change slowly, or lower it if a point genuinely needs faster command response.

- **Poll Scheduler** – specifies polling rates and displays network communication metrics:
  - Fast Rate: 10 seconds
  - Normal Rate: 30 seconds
  - Slow Rate: 1 minute

In most cases only **License** and **Serial Config / Port Name** values need to be changed.

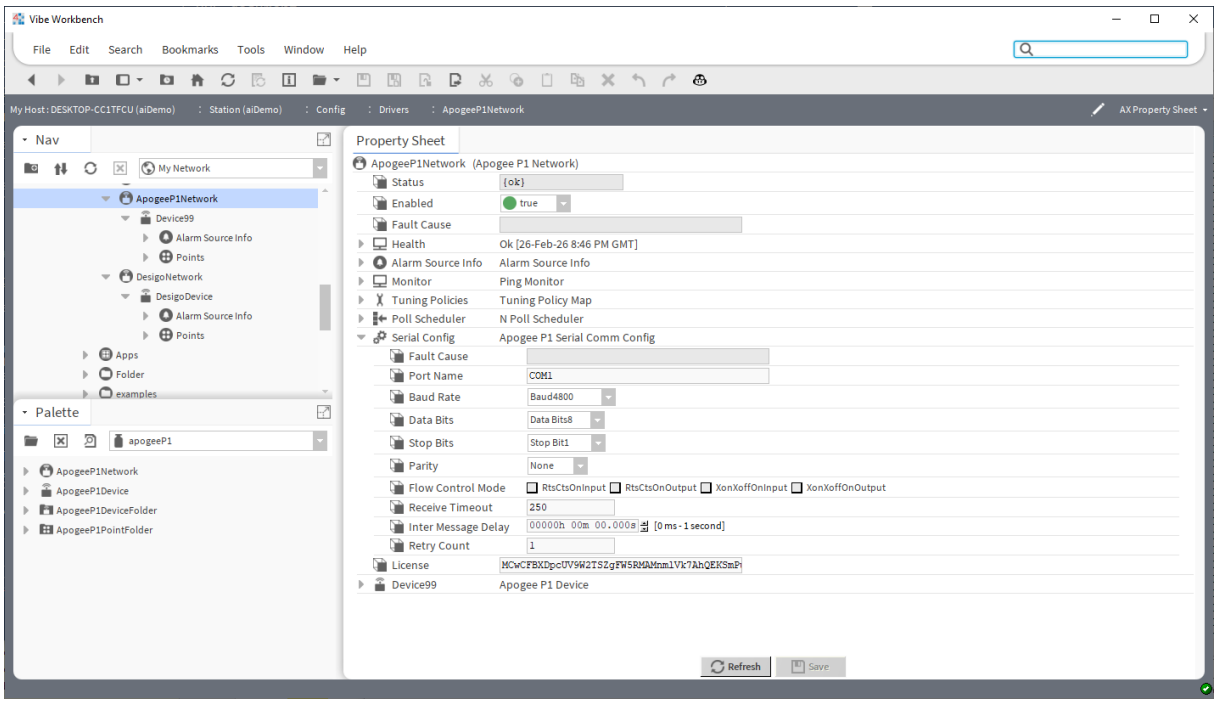

Figure 1: Apogee P1 Network property sheet

After network properties are set, use **Apogee P1 Device Manager** view to start device discovery. The driver scans all addresses from 1 to 99 and pings each one. Devices that respond are listed for addition.

5 Apogee P1 Devices

After the device discovery, the top pane displays all found devices. Each device is displayed with address, firmware revision, and application code. Devices are added to Niagara by dragging from the top pane or by selecting and pressing **Add** button.

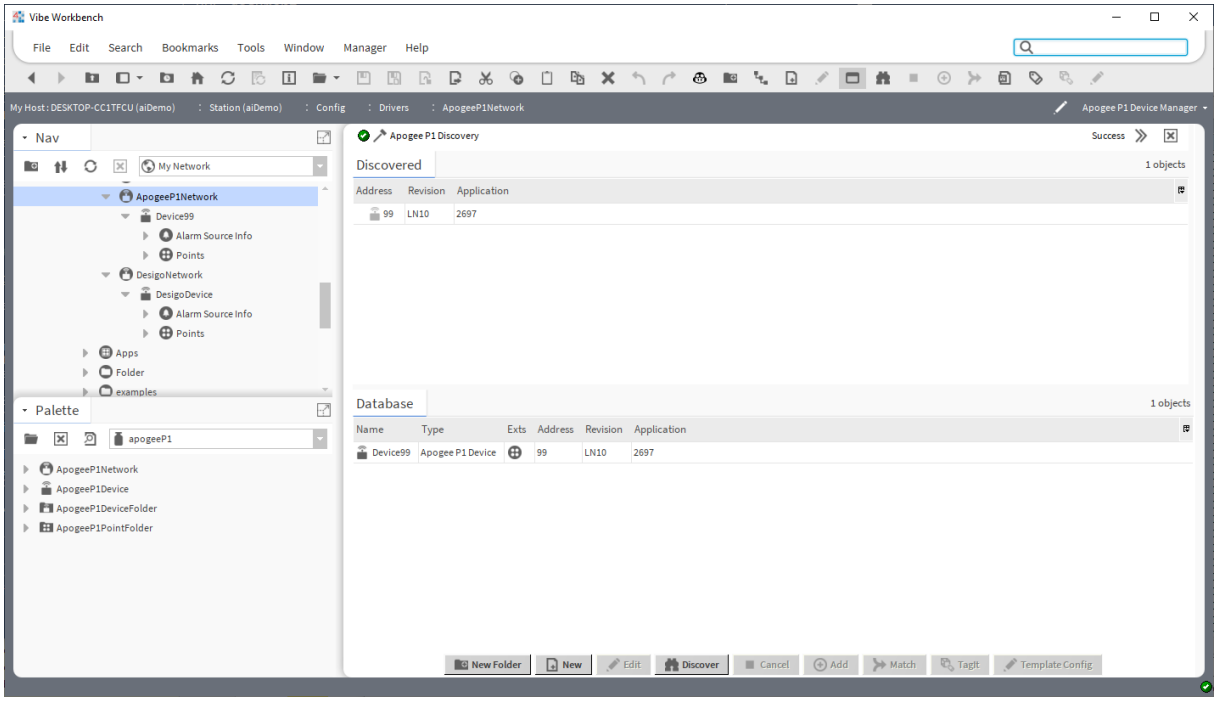

Figure 2: Apogee P1 Device Manager with discovered device

Device properties:

- **Address** – P1 network address (1-99).
- **Revision** – firmware revision string read from the device (read-only).
- **Application** – application code identifying the device type and point configuration (read-only).
- **Points** – point device extension for managing points and running point discovery.

The **Ping** action tests communication with the device and updates **Revision** and **Application** properties. If the device does not respond, its status changes to **down** and the **Health** property records the failure – **Last Fail Time** and **Last Fail Cause**, with **Last Ok Time** showing when the device last answered. Health is the quickest way to tell a device that has never communicated from one that dropped out later.

6 Apogee P1 Points

Points are discovered in **Apogee P1 Point Manager** view. The driver sends an Upload Point query to each point number (1 to 99) and retrieves point metadata directly from the device. Each discovered point has:

- **Point Number** – P1 point number (1-99).

- **Point Name** – name read from the device.
- **Point Type** – type code (LAI, LAO, LDI, LDO, etc.).
- **Value** – current real-time value.
- **Unit** – engineering unit string from the device.
- **Slope** – scaling multiplier.
- **Intercept** – scaling offset.
- **False Text** / **True Text** – state labels for boolean points (e.g., “Off” / “On”).

Points are added to Niagara by dragging from the top pane or by selecting and pressing **Add** button. The driver automatically determines the correct Niagara point type (Numeric Point, Numeric Writable, Boolean Point, or Boolean Writable) based on the P1 point type.

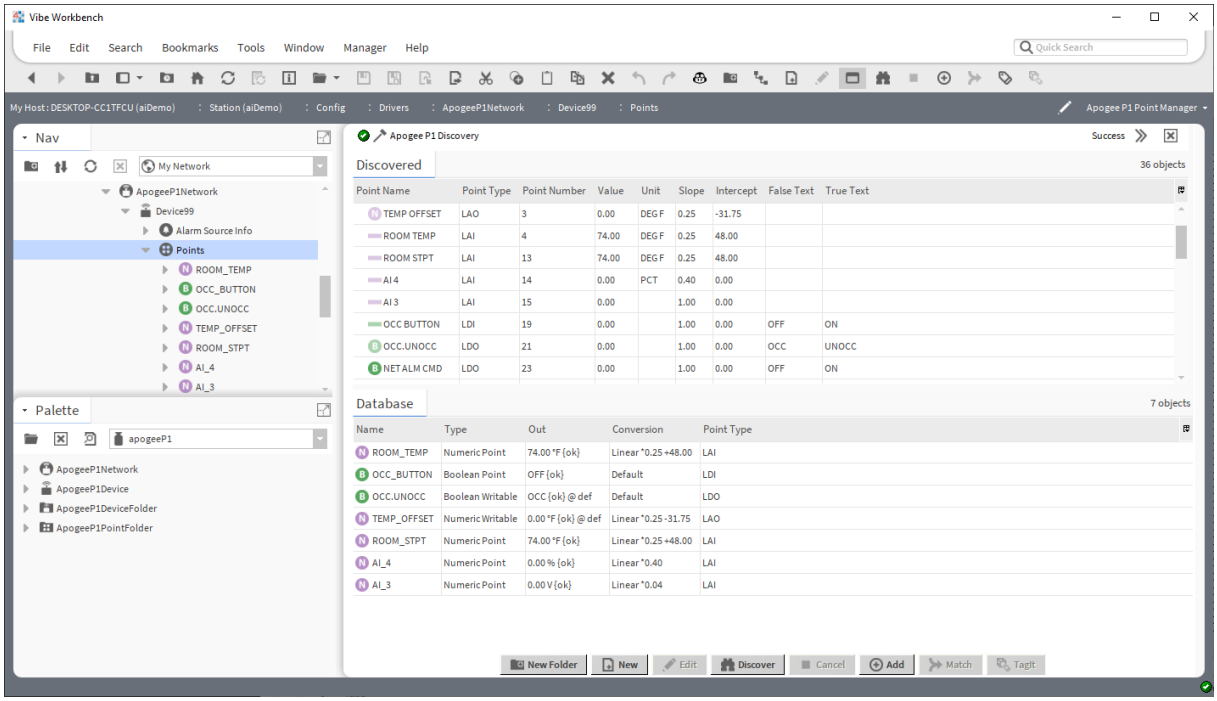

Figure 3: Apogee P1 Point Manager with discovered and added points

6.1 Point Types

Type	Code	Direction	Data Type
LAI	1	Read-only	Numeric
LAO	2	Read/Write	Numeric

Type	Code	Direction	Data Type
LPACI	3	Read-only	Numeric (Pulse Accumulator)
LDI	4	Read-only	Boolean
LDO	5	Read/Write	Boolean
L2SL	6	Read/Write	Boolean (2-State Latched)
L2SP	7	Read/Write	Boolean (2-State Pulsed)
LOOAL	8	Read-only	Numeric (On/Off/Auto)
LOOAP	9	Read/Write	Numeric (On/Off/Auto)
LFSSL	10	Read/Write	Boolean (Fast/Slow/Stop)
LFSSP	11	Read/Write	Boolean (Fast/Slow/Stop)
LENUM	12	Read/Write	Enumerated

6.2 Point Facets

When points are added to Niagara, the driver automatically assigns facets based on point metadata:

- **Engineering units** – the P1 unit string (e.g., “DEGF”, “CFM”, “PSI”) is mapped to the corresponding Niagara unit. Supported unit categories include temperature, pressure, flow, power, energy, electrical, speed, volume, length, area, time, frequency, and air quality.
- **Boolean state labels** – for boolean point types (LDI, LDO, etc.), the **True Text** and **False Text** labels from the device are assigned as Niagara boolean facets (e.g., “On” / “Off”, “Open” / “Closed”).
- **Unit conversion** – the driver sets **device facets** with the unit reported by the device (e.g., Fahrenheit). The user can change the **point facets** to a different unit (e.g., Celsius), and Niagara automatically converts the displayed value.
- **Precision** – numeric points are displayed with one decimal place by default.

6.3 Value Scaling

Numeric point values are converted between raw protocol values and engineering units using linear conversion:

```
Engineering Value = (Raw Value * Slope) + Intercept
Raw Value = (Engineering Value - Intercept) / Slope
```

The slope and intercept values are read from the device during point discovery and automatically configured on the proxy extension. No manual configuration is required.

6.4 Proxy Extension Properties

Each added point contains an **Apogee P1 Proxy Ext** with the following properties:

- **Point Number** – P1 point number (1-99). Set during discovery.
- **Point Type** – P1 point type (LAI, LAO, etc.). Set during discovery.
- **Scaling** – linear conversion with slope and intercept for raw-to-engineering value conversion. Set during discovery.
- **Tuning Policy** – assigns a tuning policy that controls poll frequency and read-after-write behavior. Select from the policies defined on the network.

7 Writing and Releasing Points

7.1 Writing Values

For writable points (LAO, LDO, L2SL, L2SP, LOOAP, LFSSL, LFSSP, LENUM), write to the Niagara control point. The driver converts the engineering value back to raw protocol format using the inverse of the linear conversion and sends it to the device.

EEPROM wear warning: stored values live in device memory that accepts a limited number of writes before it wears out, and that budget belongs to each stored value rather than to the device as a whole. Writes are not stored by default. Before enabling storage on a point, check how often it is actually written – a point driven by a control loop, a schedule, or any other logic can exhaust the budget in days and permanently damage the controller. Loop outputs are the clearest case to avoid: the controller recalculates them at startup, so nothing is gained by storing them. Keep storage for values that change occasionally and must survive a power cycle.

7.2 Releasing Overrides

To release an override on a point, write a null value to the point. The driver sends a P1 Release command to restore the point to its program-controlled value.

8 FAQ

8.1 A device appears twice in discovery

Address 99 is a broadcast address – all TEC controllers respond to it regardless of their configured address. If a single device is connected, it will appear once at its configured address and once at address 99. Only add the device at its configured address, not at 99.

Warning: Never write to a device added at address 99 on a trunk with more than one controller. A write to address 99 is a broadcast – every controller on the trunk receives and executes it. In particular, writing point 1 (CTLR ADDRESS) at address 99 changes the address of every controller on the trunk at once, leaving all controllers with the same address and the trunk unusable until each controller is disconnected and re-addressed individually. Use address 99 only on a bench with a single device connected, e.g., to assign an address to a factory-fresh controller.

8.2 I cannot discover any devices

- Check RS-485 wiring and termination resistors at both ends of the bus.
- Verify the COM port name is correct and the port is not in use by another application.
- Ensure devices are powered and have valid unique addresses.
- Verify baud rate matches the devices on the bus (4800 bps for most P1 devices).
- Check if the license is valid. Without a valid license, the network shows a configuration fault and does not send any requests.

8.3 Points are not reading

- Verify the device is online (check device status – should show **ok**).
- If the device shows **down**, run the **Ping** action to test communication.
- Check that the point number and point type match the actual device configuration.
- Increase **Receive Timeout** if the network has high latency.

8.4 A value I wrote is lost after the controller is power cycled

By default a write takes effect immediately but is not stored, so the controller loses it on restart. Assign the point the **Persistent Write Policy** to have written values stored as well. Read the EEPROM wear warning above first – do this only for points written occasionally, never for anything driven by automatic logic.

8.5 Points intermittently go to fault on a busy trunk

Only one device can use the trunk at a time, and at 4800 bps each exchange occupies it for roughly 20ms. A point written continuously by a Niagara control loop can take up most of that capacity and starve the poll cycle, leaving unrelated points to time out and drop to fault. Check the **Poll Scheduler** statistics for the actual transaction rate. Give frequently written points their own policy with a higher **Min Write Time** and **Read After Write** turned off – together these cut the traffic such a point generates several times over. Increasing **Receive Timeout** or **Retry Count** treats the symptom rather than the cause.

8.6 Communication errors or CRC errors

- Check RS-485 wiring and connections. Poor wiring is the most common cause of CRC errors.
- Verify proper RS-485 termination resistors at both ends of the bus.
- Check for electrical interference on the RS-485 bus.
- Verify baud rate matches on all devices and in the driver configuration.
- Increase **Retry Count** if errors are intermittent.