Niagara 4 Siemens Desigo CC Driver Guide

Technical Document

baudrate.io

niagara4

July 13, 2025

Contents

1 Introduction **2**

2 Requirements **3**

3 Desigo CC Server Configuration **3**

4 Quick Start **3**

5 Desigo Networks **4**

6 Desigo Devices **6**

7 Desigo Points **7**

8 Tuning Policies **9**

9 Authentication and Security **9**

1 Introduction

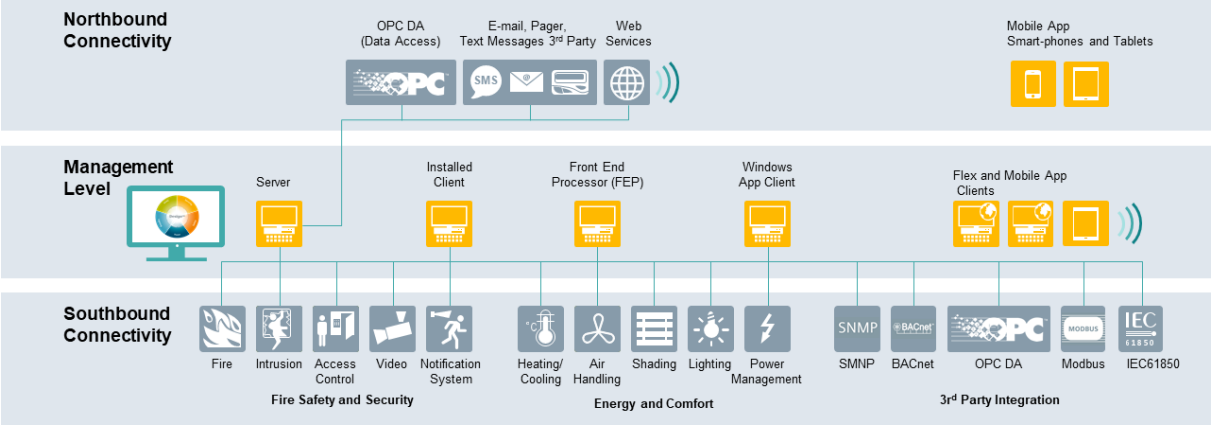

Figure 1: Desigo architecture

Desigo CC is a comprehensive and scalable building management platform from Siemens that consolidates HVAC, lighting, fire safety, security, and energy systems into a unified interface [1]. Built to serve mission-critical facilities such as large-scale campuses, airports, and hospitals, Desigo CC offers deep integration capabilities, visual supervision, and rich data services.

The Desigo CC system provides:

- **Centralized Building Management** - Unified platform for all building automation disciplines including HVAC, lighting, security, and fire safety systems.
- **Enterprise Integration** - Northbound Open RESTful Integration Service (NORIS) for seamless connectivity with third-party systems and cloud platforms.
- **Scalable Architecture** - Supports small facilities to large enterprise deployments with distributed server configurations.
- **Web-based Operations** - Modern web interface for system visualization, control, and management.
- **Standards Compliance** - Built on open standards including BACnet, OPC, and RESTful web services.

Desigo CC driver for Tridium Niagara uses the Northbound Open RESTful Integration Service (NORIS) to communicate directly with the Desigo CC server via secure HTTP/HTTPS protocols. The driver leverages OAuth 2.0 token-based authentication and supports multi-object request-response for efficient data exchange.

[1] Siemens Desigo CC is a trademark of Siemens AG. All other trademarks are property of their respective owners.

The driver automatically discovers all system objects and properties across all disciplines, exposing them as standard Niagara points. This enables real-time reading, writing, and command execution while maintaining access rights and security protocols established in the Desigo CC system.

2 Requirements

- Niagara framework host such as Supervisor, JACE or any other Niagara-compatible device
- Desigo CC driver module and license
- Access to Desigo CC server with NORIS interface enabled
- Valid user credentials with appropriate permissions in Desigo CC
- Network connectivity between Niagara and Desigo CC server

3 Desigo CC Server Configuration

Before using the driver, ensure that the Desigo CC server has the **Northbound Open RESTful Integration Service (NORIS)** enabled and properly configured. This is typically configured through the Desigo CC system administration interface.

The Desigo CC server must have:

- **NORIS interface enabled** - Provides RESTful API access for third-party integrations
- **Valid user account** - Username and password with appropriate system access rights
- **OAuth 2.0 authentication** - Token-based security for API access
- **Network connectivity** - Ensure firewall settings allow HTTPS traffic to NORIS endpoints
- **Access rights configured** - User permissions must allow access to required system objects and operations

4 Quick Start

1. Copy desigo-rt.jar module to Niagara /modules folder and restart the software.
    a. If the driver is hosted on PC: restart Supervisor station and Workplace.
    b. If the driver is hosted on JACE: import the module with all its dependencies into JACE and restart it.
2. Add **Desigo Network** to the station.
3. Enter license code in **License** property under **Desigo Network**.

4. Add **Desigo Device** under the network.
5. Enter **Url** (e.g., `https://desigo-cc-server/WebServices/`) in the device properties.
6. Configure **Auth** with valid username and password for the Desigo CC system.
7. Test connection using the **Login** action on the device to verify OAuth 2.0 authentication.
8. Open device **Points** extension and **Discover** system objects across all disciplines.
9. Add discovered objects to the station. Objects can be configured for reading, writing, and command execution as needed.

5 Desigo Networks

Desigo Network contains standard Niagara properties plus driver-specific configurations:

- **License** – the code which allows the driver to run on your host.
- **Http Config** – HTTPS communication settings including timeouts.
- **Poll Scheduler** – specifies polling rates and displays network communication metrics for monitoring NORIS API performance and health.
- **Tuning Policies** – see Tuning Policies section for detailed polling configuration.

The network automatically handles OAuth 2.0 token refresh and secure HTTPS communication with the Desigo CC server. In most cases, only the **License** property needs to be configured at the network level.

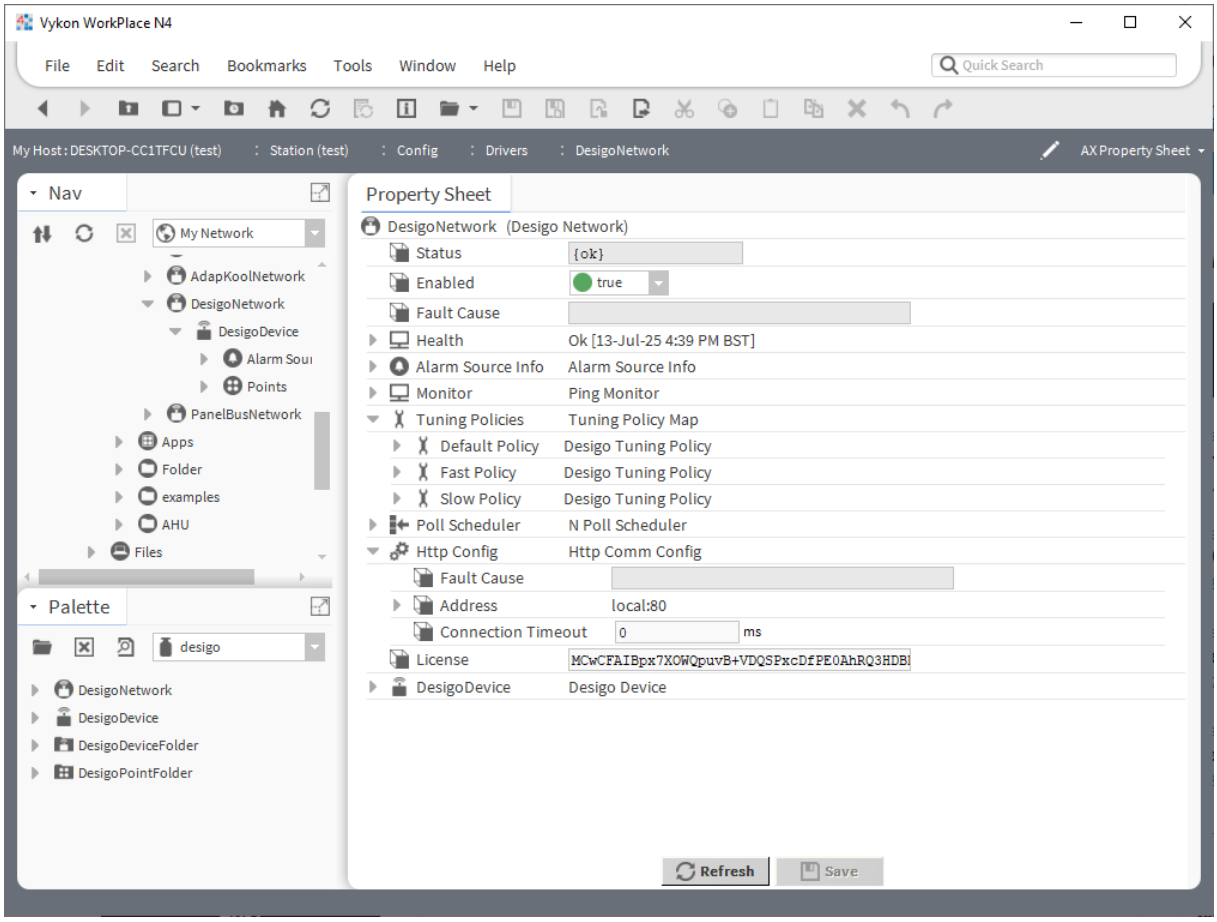

Figure 2: Desigo Network properties

6 Desigo Devices

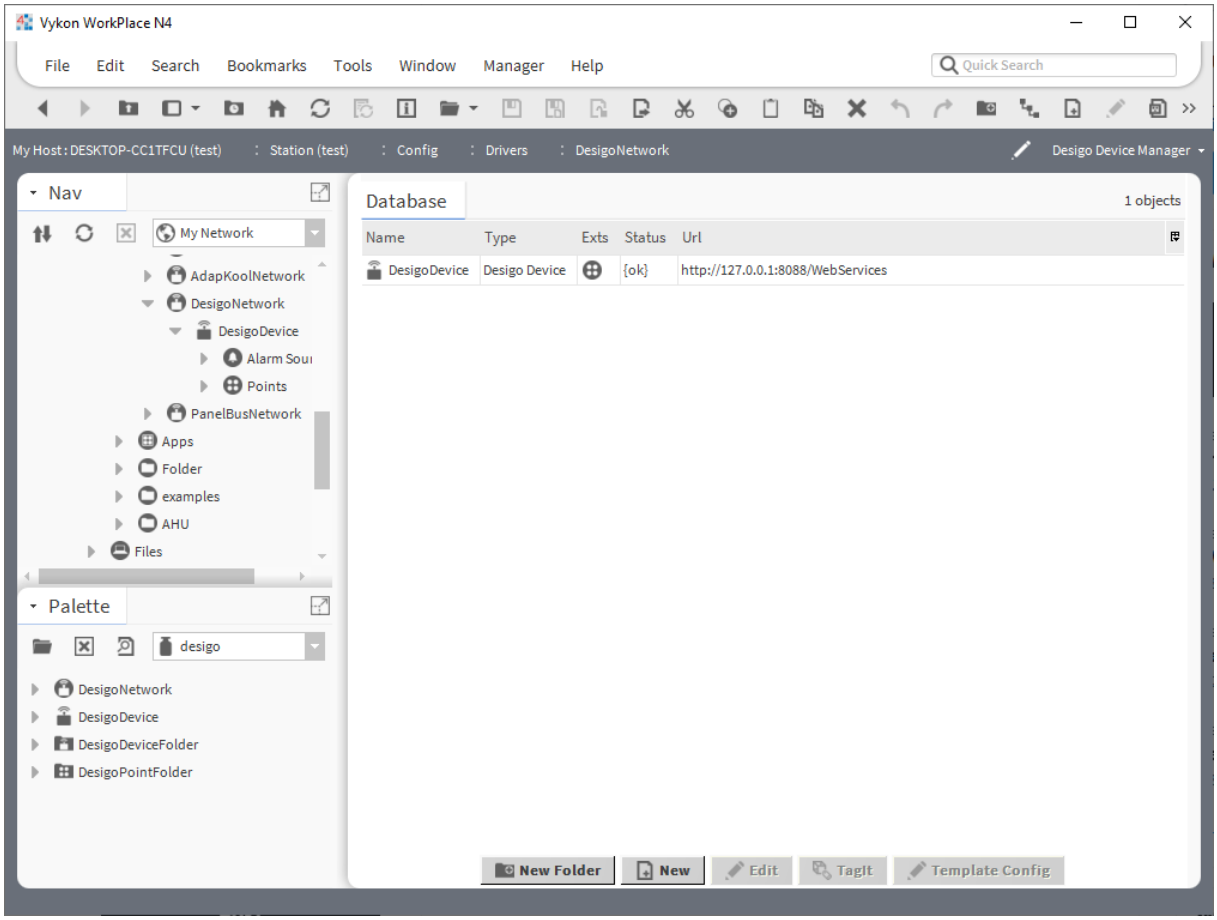

Figure 3: Desigo devices

Desigo CC Device represents a connection to the Desigo CC server and contains the following key properties:

- **Url** – Complete NORIS endpoint URL for the Desigo CC server (e.g., `https://desigo-cc-server/WebServices/`).
- **Auth** – Username and password credentials for Desigo CC system authentication.
- **Point Group Size** – Maximum number of points per bulk request for optimized multi-object operations (default: 100).
- **Current Number Of Groups** – Read-only display of active polling groups across all disciplines.
- **Points** – Point management extension for discovery and organization.

The device automatically validates the NORIS API availability and tests OAuth 2.0 authentication when properties are changed. Use the **Login** action to verify credentials and server connectivity.

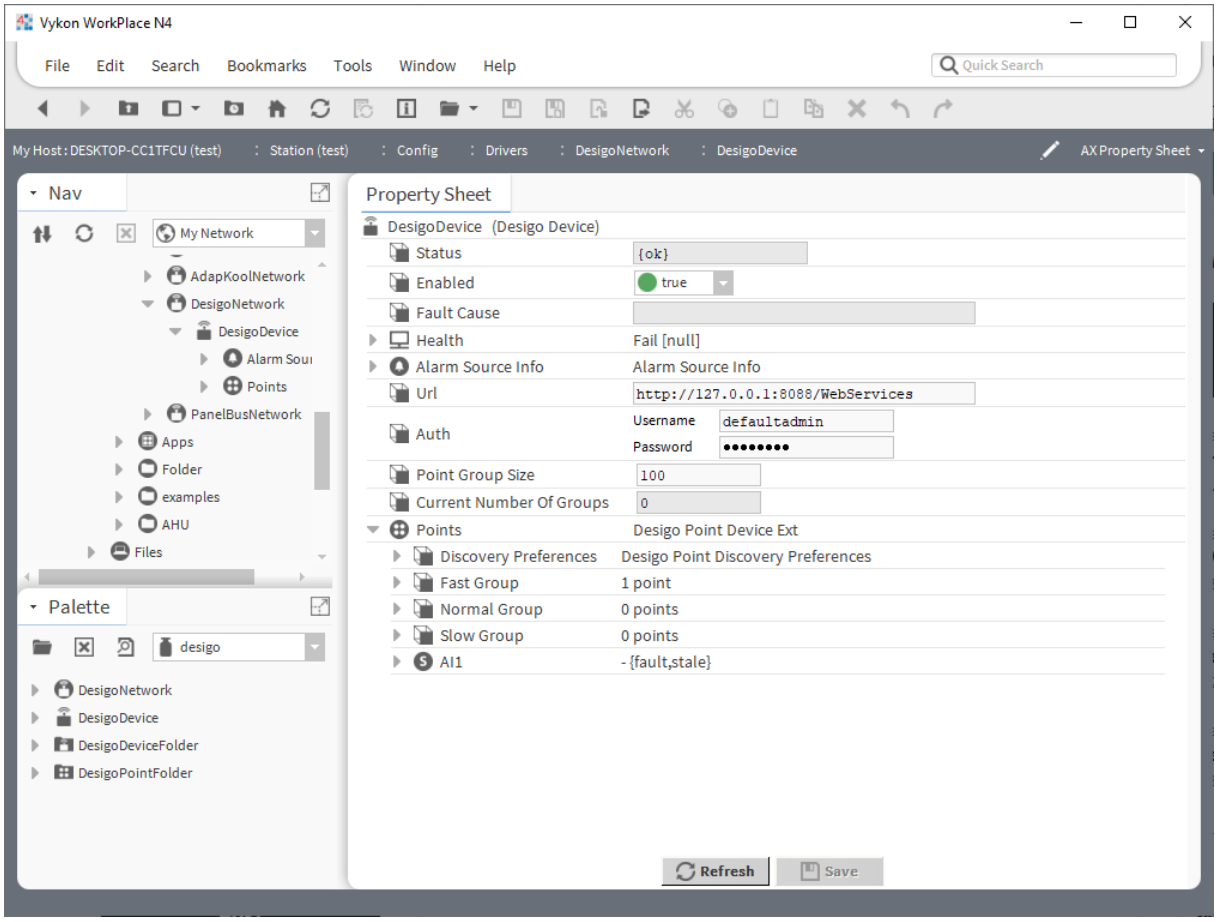

Figure 4: Desigo device properties

7 Desigo Points

Desigo CC objects are discovered through the NORIS API interface. The discovery process automatically traverses the system's object hierarchy across all disciplines and identifies all available data points and system objects.

Each discovered object displays:

- **Object Name** – The object name as configured in the Desigo CC system.
- **Object Id** – Unique identifier used for NORIS API communication.
- **Data Type** – Object data type (Boolean, Numeric, String, Enum, etc.).
- **Value** – Current real-time object value.
- **Unit** – Physical units (when available).

Objects are organized in a folder structure that mirrors the Desigo CC system hierarchy across all disciplines, making navigation intuitive for users familiar with the system organization.

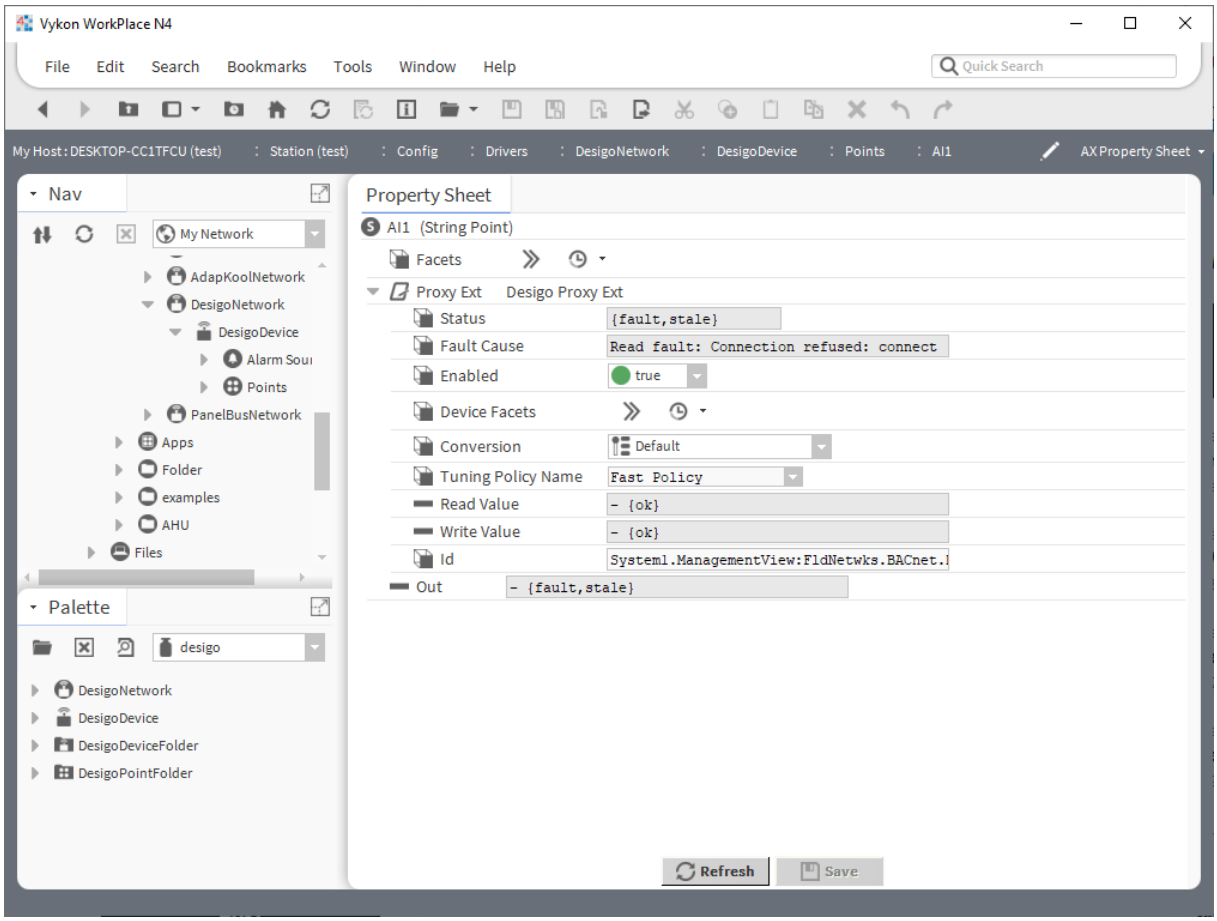

Figure 5: Desigo point

When adding objects to Niagara:

- **Name** – Display name for the object in Niagara (can be customized).
- **Type** – Niagara point type automatically determined from Desigo CC data type and could not be changed after creation.
- **Tuning Policy Name** – Assigns polling frequency from available tuning policies.
- **Id** – Desigo CC object reference

The driver automatically maps Desigo CC data types to appropriate Niagara point types and applies correct units and precision based on the object configuration. Access rights from Desigo CC are preserved to ensure security compliance.

8 Tuning Policies

Tuning Policies define how objects are read and written, with each policy containing:

- **Min Write Time** – Minimum time between writes for writable objects to prevent rapid value changes.
- **Max Write Time** – Maximum wait time before rewriting values to ensure updates reach the Desigo CC server.
- **Write On Start** – Controls whether writable objects are updated when the station starts.
- **Write On Up** – Determines if objects are written when the device comes online.
- **Write On Enabled** – Controls writing behavior when objects transition from disabled to enabled.
- **Stale Time** – Time after which objects become "stale" if no successful reads occur.
- **Poll Frequency** – Associates the policy with Fast Rate, Normal Rate, or Slow Rate polling.
- **First Poll** – Whether the point shall be polled as soon as possible after Niagara designates it as **subscribed**. Shall be set to *true* for classical BMS application and to *false* for gateway-type applications.

Initially, three policies are created:

- **Default Policy** – Normal rate polling (typically 10 seconds)
- **Fast Policy** – Fast rate polling (typically 5 seconds)
- **Slow Policy** – Slow rate polling (typically 30 seconds)

Objects are automatically assigned to polling groups based on their tuning policy, with the driver performing bulk NORIS API requests for efficiency. Each group can contain up to 100 objects (configurable via Point Group Size) and supports multi-object request-response for optimal performance.

9 Authentication and Security

The Desigo CC driver implements enterprise-grade security using:

- **OAuth 2.0 Authentication** – Industry-standard token-based authentication for secure API access.
- **HTTPS Communication** – All communication with NORIS API uses TLS encryption.
- **Token Management** – Automatic token refresh and session management.
- **Access Rights Preservation** – Respects user permissions and access rights configured in Desigo CC.
- **Secure Credential Storage** – Passwords are protected using Niagara's security framework.

The driver automatically handles OAuth 2.0 token lifecycle including initial authentication, token refresh, and re-authentication when needed. All communication with the Desigo CC server follows secure web protocols and maintains the access rights established in the Desigo CC system.