

powered by

niagara  
framework



## Niagara and ILC 2050 BI – Getting started

Quick start guide

## Quick start guide

### Niagara and ILC 2050 BI – Getting started

2018-05-04

---

Designation: UM QS EN NIAGARA ILC 2050 BI

Revision: 01

Order No. —

This user manual is valid for:

Designation	As of version (HW)	As of version (FW)	Order No.
ILC 2050 BI	02	–	2403160

## Table of contents

1	Niagara Framework .....	5
1.1	Niagara host.....	5
1.2	Niagara platform.....	6
1.3	Niagara modules .....	6
1.4	Niagara station .....	6
1.5	Niagara system structure .....	7
2	Niagara Workbench.....	8
2.1	Niagara Supervisor installation.....	8
2.2	Installing SysMik Niagara AddOns .....	9
2.3	Niagara Supervisor file system.....	9
2.4	Niagara Supervisor licensing.....	9
2.5	System passphrase.....	9
2.6	Starting the Niagara Workbench .....	10
2.7	Help.....	10
3	ILC 2050 BI .....	12
3.1	Hardware installation.....	12
3.1.1	Power supply .....	12
3.1.2	Network connection .....	12
3.1.3	USB port.....	13
3.2	Commissioning tools .....	13
3.2.1	Commissioning website .....	13
3.2.2	SCA system shell .....	14
4	Niagara commissioning .....	15
4.1	Preparation.....	15
4.1.1	Niagara version .....	15
4.1.2	License .....	15
4.1.3	New station.....	15
4.1.4	Establishing the platform connection .....	17
4.2	Configuring commissioning steps.....	20
4.2.1	Licensing .....	21
4.2.2	Runtime profiles.....	22
4.2.3	Installing a station.....	23
4.2.4	Installing modules.....	24
4.2.5	Installing the system software.....	25
4.2.6	Setting the time.....	25
4.2.7	IP settings.....	26

**Table of contents**

	4.2.8	Configuring platform users .....	27
	4.3	Completing commissioning .....	28
<b>5</b>		<b>Station engineering.....</b>	<b>30</b>
	5.1	Components.....	30
	5.2	Views .....	30
	5.3	Station connection.....	31
	5.4	Hello World – first application.....	33
	5.4.1	Installing the module.....	33
	5.4.2	Creating the component .....	34
	5.4.3	Configuring the component .....	35
	5.4.4	Connecting components - links.....	36
	5.5	Points .....	37
	5.5.1	Point value and status .....	37
	5.5.2	Writable points.....	37
	5.5.3	Read-only points.....	38
	5.5.4	Facets.....	38
	5.6	Trend data collection - history .....	39
	5.7	Alarm management.....	43
	5.8	Scheduling .....	46
	5.9	Web visualization .....	47
	5.9.1	User-defined views .....	47
	5.9.2	User management.....	54
	5.10	Drivers.....	58
	5.10.1	I/O driver - SysmikScaloNetwork.....	59
	5.10.2	Serial drivers.....	62
	5.11	Saving a station.....	63
	5.11.1	Saving a station .....	63
	5.11.2	Copying a station.....	63
	5.11.3	Backup .....	64
	5.12	Application structure.....	64
<b>6</b>		<b>Niagara Network.....</b>	<b>65</b>
	6.1	Niagara Supervisor station.....	65
	6.2	Communication .....	66
	6.3	Integrating a station.....	66
	6.3.1	Fox .....	66
	6.3.2	Foxs.....	66
	6.4	Integrating points into the Supervisor Station.....	68
	6.5	Integrating trend data into the Supervisor Station .....	69
	6.6	Integrating schedules into the Supervisor Station .....	69

**Table of contents**


---

6.7	Central alarms .....	69
<b>7</b>	<b>Security .....</b>	<b>70</b>
7.1	Passwords.....	70
7.2	Login credentials .....	70
7.3	Protocols .....	70
7.4	Passphrase .....	70
7.5	Environment.....	70
7.6	ILC 2050 BI .....	70
<b>8</b>	<b>Niagara diagnostics .....</b>	<b>71</b>
8.1	Version information .....	71
8.2	Event logging .....	71
8.2.1	Niagara station .....	71
8.2.2	Niagara Workbench.....	72
8.3	Execution of applications .....	72
8.4	System diagnostics - Spy.....	73
8.4.1	EngineManager .....	73
8.4.2	ClassLoaders.....	73
8.4.3	Log setup.....	73
<b>9</b>	<b>Software maintenance.....</b>	<b>74</b>
9.1	Niagara Workbench .....	74
9.2	Supervisor.....	74
9.3	IB IL 2050 BI .....	74

**Table of contents**

---

# 1 Niagara Framework

The Niagara Framework is a universal software kit for automation systems.

It has a highly modular structure and provides ready-to-use functions for many standard requirements. The functionality can be extended in almost every imaginable direction by programming additional modules in JAVA.

Standard functions include:

- A **Niagara Workbench** engineering tool for all levels of the system
- Integration of information from lower-level third-party systems via numerous communication drivers
- Transfer of information to higher-level third-party systems via numerous communication drivers
- Data processing
- Alarm management
- Scheduling
- Trend data collection
- Process visualization
- Reports
- Data analysis
- E-mailing / text messaging

Typical software extensions include special communication drivers as well as industry-specific modules and customizations. The Niagara Community is comprised of hardware and software manufacturers for the Niagara environment as well as users, system integrators and training centers [www.niagara-community.com](http://www.niagara-community.com). This community offers products and services for the Niagara Framework.

## 1.1 Niagara host

Niagara software runs on different hardware, called the **Niagara host** in the Niagara environment. Niagara software ships with different Niagara controllers, such as the **ILC 2050 BI**. There is also Niagara software product called the **Niagara Supervisor**, which can be installed on computers with Windows or Linux operating systems, from simple PCs to virtual machines in the cloud.

## 1.2 Niagara platform

A hardware- and OS-specific software application provides the **Niagara platform** on every Niagara host. The platform is basically an operating system extension that provides a platform-independent Niagara environment. Using the Niagara platform, hosts can be centrally managed in the Niagara Workbench, the engineering tool of the Niagara Framework. This includes:

- IP settings
- Software updates
- Installing, starting, stopping applications, creating backups
- License and certificate management

If a Niagara Supervisor for Windows operating systems is used, the Niagara platform consists of a Windows service called **Niagara**. It is also called the platform daemon.

## 1.3 Niagara modules

Niagara has a modular structure. Niagara modules are units of software for distinct groups of functions. These Niagara modules contain components that implement a specific function. They are very similar to libraries and corresponding function blocks in PLC programming.

Examples of modules include communication drivers that implement a communication protocol.

## 1.4 Niagara station

In the engineering process, the Niagara Workbench is used to assemble an application from numerous components of various modules. This application is called Niagara station.

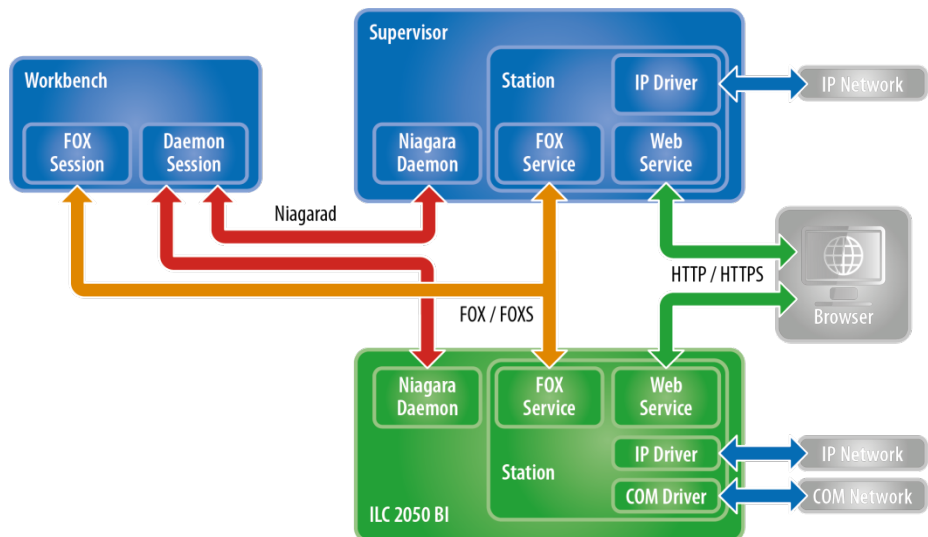


Figure 1-1 Niagara station – engineering

## 1.5 Niagara system structure

Niagara Stations can communicate with each other. In typical system structures, Niagara controllers, which serve as integration platforms in the field, perform a variety of functions:

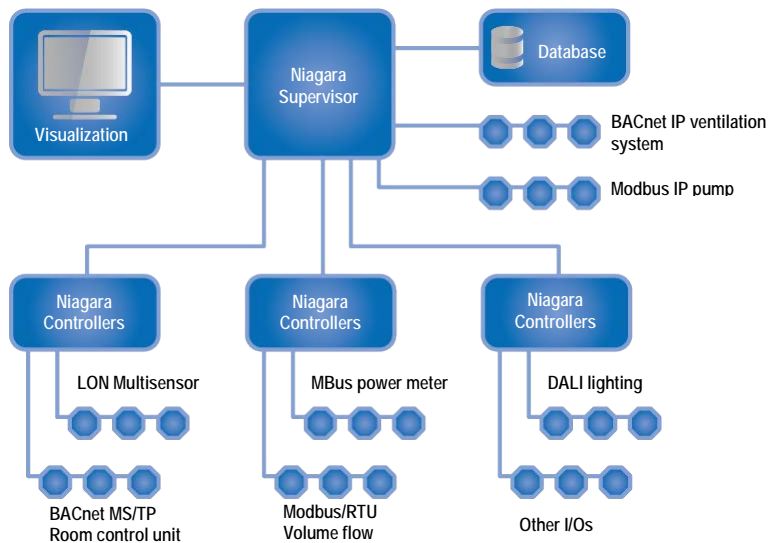
- Record and normalize field level data via serial or IP communication drivers
- Run local control algorithms
- Generate alarms
- Run schedules
- Record trend data

Several of these embedded stations may be connected to a central supervisor station, which performs the following tasks:

- Integrate data from Niagara controllers
- Access smart devices via IP communication drivers
- Run central control algorithms
- Provide web-based process visualization
- Consolidate all alarms in an alarm console, alarm management
- Synchronize schedules of Niagara controllers
- Archive trend data of all Niagara controllers and export them to external databases, if necessary
- Central user management for the entire Niagara system
- Central management of software updates
- Regular backups of all Niagara stations.

Depending on the size of the automation system, additional hierarchy levels (buildings – property – country – continent – world) can be mapped in a Niagara system.

Figure 1-2 Example of Niagara system structure



## 2 Niagara Workbench

The Niagara Workbench is the engineering tool in the Niagara Framework. You can use the Niagara Workbench to manage all Niagara hosts via the Niagara platform and program Niagara stations (applications).

The Niagara Workbench is not available as a separate product. Instead, it is part of every Niagara supervisor, installed with the Niagara Supervisor installation, and licensed with the Niagara Supervisor license.

### 2.1 Niagara Supervisor installation

You can install the Niagara Supervisor from a DVD, or you can download the contents of the DVD as a .zip file under Downloads on the product page of the ILC 2050 BI controller at [www.phoenixcontact.com](http://www.phoenixcontact.com). The Installation Wizard opens when you run the **Start.exe** file. Once you have confirmed the mandatory end user license agreement, you can select an installation directory (default: C:\niagara). In the subsequent dialog boxes, you can select and launch the version you want to install (32-bit Niagara Supervisor or 64-bit Niagara Supervisor).



#### 32 bit or 64 bit

Two different versions of Niagara Supervisor for Windows can be installed. The 32-bit Niagara Supervisor uses less memory but can address a maximum of 1.5 GB of RAM. The 64-bit Niagara Supervisor can eliminate this limitation if a Niagara Supervisor station requires more memory (if many Niagara controllers or third-party systems are integrated).

All licenses are based on host IDs. The host ID for the Niagara supervisor is determined from the hardware and software characteristics of the host. The host ID of a host is different for 32-bit Niagara Supervisors and 64-bit Niagara Supervisors. Therefore, for licensing reasons, only 32-bit Niagara Supervisor versions or only 64-bit Niagara Supervisor versions should be installed on a machine. This will ensure that all installed Niagara Supervisor versions can run on a single Niagara Supervisor license.

You can switch between versions by transferring the license from the original host ID to the new host ID.

Multiple versions of the Niagara Supervisor (including Niagara Workbench) can be installed on one computer. Whereas only one platform daemon (Windows service) can be active, all installed versions of the Niagara Workbench can run concurrently.

Newer versions of the Niagara Supervisor are usually always installed in dedicated installation folders. This allows you to continue using older versions without restrictions.

## 2.2 Installing SysMik Niagara AddOns

If you want to use the Niagara Workbench of an installed Niagara Supervisor for engineering Niagara ILC 2050 BI controllers, you must also install SysMik Niagara AddOns in the installation directory of the Niagara supervisor. SysMik Niagara AddOns include Niagara modules needed to run the ILC 2050 BI controller and up-to-date firmware for the ILC 2050 BI controllers. SysMik Niagara AddOns are always related to a specific Niagara minor version (e.g. Niagara 4.2) and can only be installed in corresponding Niagara Supervisor versions (e.g. 4.2.36.38).

More recent updates to SysMik Niagara AddOns can be easily installed in an existing Niagara Supervisor installation with older, already installed SysMik Niagara AddOns. All modules and firmware files are versioned and managed in the Niagara Supervisor software database.

## 2.3 Niagara Supervisor file system

Every Niagara Supervisor installation creates a **Niagara Home** directory, for example C:\niagara\SupervisorN4-4.2.36.38. Program files of the Niagara platform and Niagara Workbench as well as Niagara modules, including documentation, are stored here.

A **Workbench User Home** link refers to the home directory of the Niagara Workbench user, for example C:\Users\\Niagara4.2\, where the data of the user of the Niagara Workbench is stored, especially stations, backups and templates.

A **Daemon User Home** link refers to the home directory of the Niagara platform, for example C:\ProgramData\Niagara4.2\, where the Niagara Supervisor stations are stored.

## 2.4 Niagara Supervisor licensing

The Niagara Workbench can be started once the Niagara Supervisor installation is completed. If the Niagara supervisor is not yet licensed, a message is displayed with the currently calculated host ID. You can now purchase a Niagara Supervisor license via the sales channel by specifying this host ID. Once generated, this license will be available on a license server and will be retrieved the next time you start the Niagara Workbench when connected to the Internet. Alternatively, you can obtain \*.license files via e-mail and copy them to the installation directory of the Niagara Supervisor, <Niagara Home>\security\licenses (typical example: "C:\niagara\SupervisorN4-4.2.36.38\security\licenses"). You can then start and begin using the Niagara Workbench.

There are multiple Niagara Supervisor licenses that differ in the number of Niagara stations that can be integrated into the Niagara Supervisor. All of these licenses allow unrestricted use of the Niagara Workbench.

## 2.5 System passphrase

As a user, when you launch a workbench for the first time on a machine, you are prompted to provide a system passphrase. This password is used to encrypt sensitive data of a station (for example, password hashes of station users) in the **Workbench User Home** directory of the workbench.



This system passphrase is important and should not be forgotten.

Niagara hosts also have a system passphrase for encryption of sensitive data. Whenever

## Niagara Workbench

a station is copied between the Workbench and host, the sensitive data is decrypted with the system passphrase of the source and immediately re-encrypted with the passphrase of the destination. If a passphrase is lost, affected stations cannot be unlocked without losing sensitive data.

## 2.6 Starting the Niagara Workbench

The Niagara Workbench can be started via the Start menu with or without the console. If certain errors occur, for example if the Niagara Workbench does not start, the console provides helpful diagnostic information. However, it is not needed for everyday use of the Niagara Workbench.

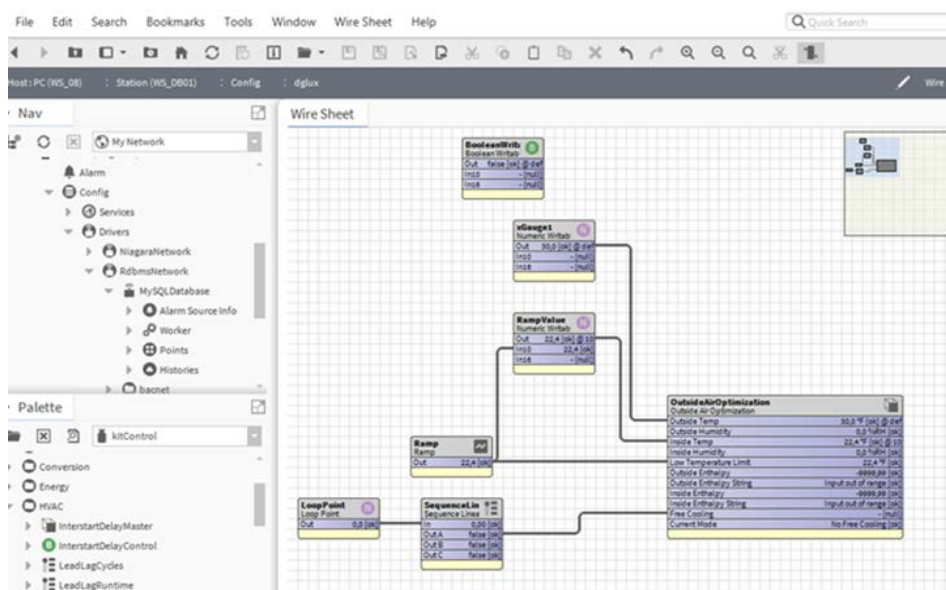


Figure 2-1 Workbench start screen

The Niagara Workbench window is divided into a central workspace and a side bar on the left. You can select the tools displayed in the side bar via the **Window – Side Bars** menu. **Nav** (navigator) is an important side bar. It contains a tree view of the Niagara system, starting with Niagara hosts, the various connections to them and the available elements (tools, files, components). It enables easy navigation: When you double-click an element in the navigation tree, the default view of this element is displayed in the workspace. Other important options are provided via the local element-specific menu.

If necessary, you can open multiple sessions of the Niagara Workbench, for example by choosing **File – New Window** from the menu. Alternatively, you can create several independent workspaces using tabs by choosing **File – New Tab** from the menu.

## 2.7 Help

The Niagara Supervisor includes a comprehensive help system. You can view the help system directly in the Niagara Workbench via the **Windows – Side Bars – Help** menu. When you open help for the first time, you must choose **LoadHelp** to load Niagara Help from the installed Niagara modules.

You can display the help system in a new **Niagara Help** window by choosing **Help – Help Contents** from the menu. From the help menu, you can also jump to specific help topics

for components or views currently displayed in the Niagara Workbench.

The help system has a modular structure. All available help modules are listed in the **Table Of Contents** tab. A good place to start is **Doc User**, which is essentially a Niagara user manual.



Alternatively, all help is also available as PDF files in the installation folder of the Niagara Supervisor under **\docs**.

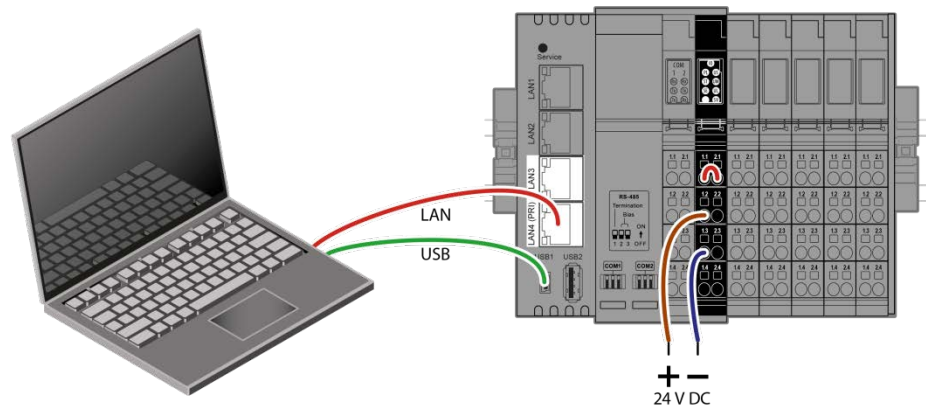
Further help can be found online on the Niagara Community website at [www.niagara-community.com](http://www.niagara-community.com). Members of the community discuss topics related to the Niagara Framework in various forums on the website. Become a member of the community and post your own questions.

## 3 ILC 2050 BI

The ILC 2050 BI is a full-fledged Niagara controller that expands the Niagara product range to include a powerful modular controller with a wide variety of alignable inline system terminals, including analog and digital inputs and outputs, DALI, M-Bus, MP-Bus, meters, serial interfaces etc. For more detailed information, see the user manual of the controller.

### 3.1 Hardware installation

The ILC 2050 BI is designed for snap-in mounting on DIN rails. At minimum, a power supply and Ethernet network are required for commissioning the ILC 2050 BI. In addition, a USB connection between the controller and PC can support the commissioning process.



### 3.1.3 USB port

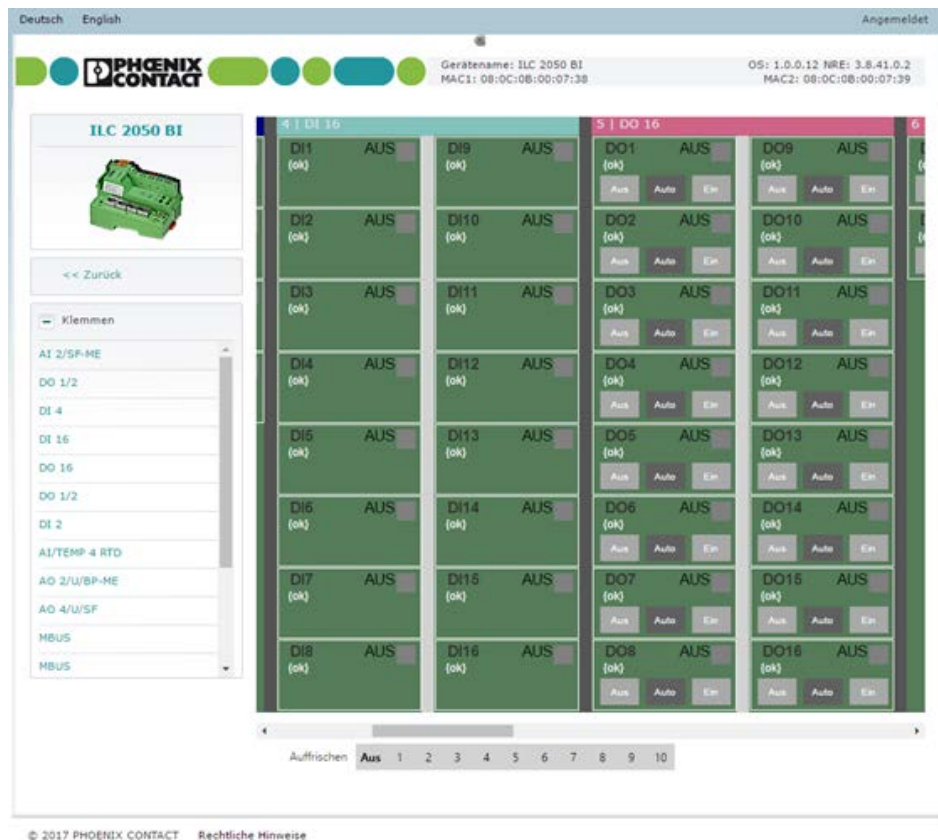
A USB cable can be used to establish a local connection between a PC and the mini USB port of the ILC 2050 BI. The Windows PC automatically installs a virtual Ethernet adapter (RNDIS) for accessing the commissioning tools of the ILC 2050 BI.

## 3.2 Commissioning tools

### 3.2.1 Commissioning website

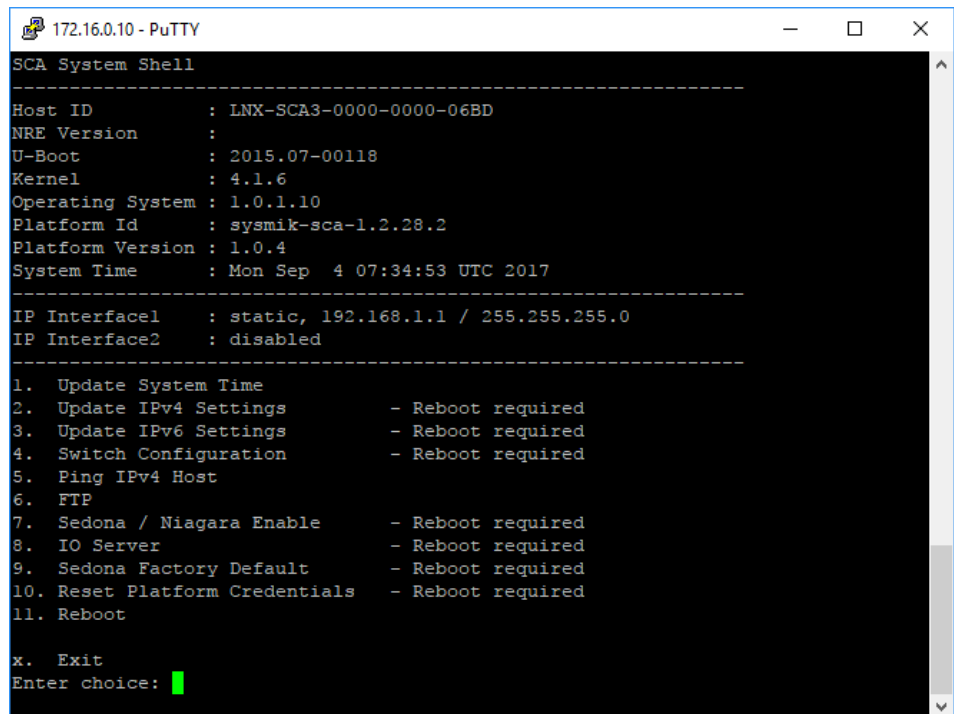
If a USB connection has been established between the ILC 2050 BI and the PC, you can open the commissioning website in a browser by entering <http://172.16.0.10:8080>. The commissioning website assists you with the wiring test for connected I/O terminals and allows you to change general settings such as IP addresses and system time.

Figure 3-2 ILC 2050 BI commissioning website



### 3.2.2 SCA system shell

If a USB connection has been established between the ILC 2050 BI and the PC, you can display and control the SCA system shell in a Telnet client via IP address **172.16.0.10**, port **23**.



```

172.16.0.10 - PuTTY
SCA System Shell
-----
Host ID       : LNX-SCA3-0000-0000-06BD
NRE Version  :
U-Boot       : 2015.07-00118
Kernel       : 4.1.6
Operating System : 1.0.1.10
Platform Id  : sysmik-sca-1.2.28.2
Platform Version : 1.0.4
System Time  : Mon Sep  4 07:34:53 UTC 2017
-----
IP Interfacel : static, 192.168.1.1 / 255.255.255.0
IP Interface2 : disabled
-----
1. Update System Time
2. Update IPv4 Settings      - Reboot required
3. Update IPv6 Settings      - Reboot required
4. Switch Configuration      - Reboot required
5. Ping IPv4 Host
6. FTP
7. Sedona / Niagara Enable   - Reboot required
8. IO Server                  - Reboot required
9. Sedona Factory Default    - Reboot required
10. Reset Platform Credentials - Reboot required
11. Reboot
x. Exit
Enter choice: █

```

Figure 3-3 System shell

As a network-enabled device, the ILC 2050 BI can be extensively configured and protected. If important information such as login details or IP addresses are lost and the controller cannot be operated, the SCA system shell provides local, fixed and unprotected access via USB. Here, platform login details can be reset at any time or IP addresses can be displayed and changed.

## 4 Niagara commissioning

Commissioning is the process of implementing the software for a Niagara host. It includes licensing, installing required software and making various system settings.

### 4.1 Preparation

Commissioning involves several steps, some of which require preparation.

#### 4.1.1 Niagara version

First, the latest version of the Niagara Workbench (2.1) as well as SysMik Niagara AddOns (2.2) must be installed. This ensures that the latest software is installed on the Niagara host during commissioning. If you want to install a specific version of Niagara on the Niagara host, the appropriate version of the Niagara Workbench should be used. This may become necessary if software maintenance has expired and the Niagara host does not have a license for the current Niagara version.

#### 4.1.2 License

To license the Niagara host, you must purchase a corresponding Niagara license via the sales channel by specifying the Host ID of the Niagara host.

#### 4.1.3 New station

In order to load a station during commissioning, you must first create it using the Niagara Workbench. Open the **New Station Wizard** by choosing **Tools – New Station** from the menu. Here you must specify the station name under which the station will be stored in the Workbench User Home directory.



The station name becomes more important when multiple stations must communicate with each other via the Niagara network. For this reason, station names must be unique throughout the network. However, the station name can be modified when stations are transferred from the Workbench to the Niagara host. In this way, a workbench station can be copied to many Niagara hosts.



You can create stations with the workbench or copy them from or to a Niagara host using the **Station Copier** platform tool. Stations can be transferred via the file system, but this method is prone to errors because version checks are skipped and sensitive data is not encrypted and decrypted by means of a passphrase.

## Niagara commissioning

You must also select a station template. The **NewControllerStation.ntpl** station template is suitable for the ILC 2050 BI.

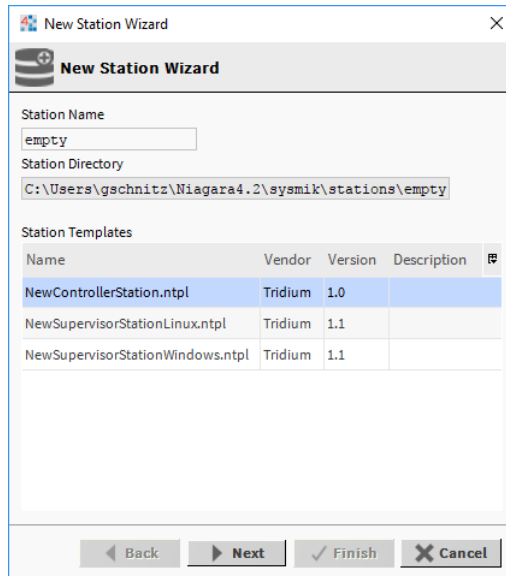


Fig 4-1 Starting the New Station Wizard

The **New Station Wizard** always assigns the station a user with the name **admin**. A password must be entered and confirmed for this user. Please use the password **Admin12345** for this exercise. These login credentials will be used later for station engineering.

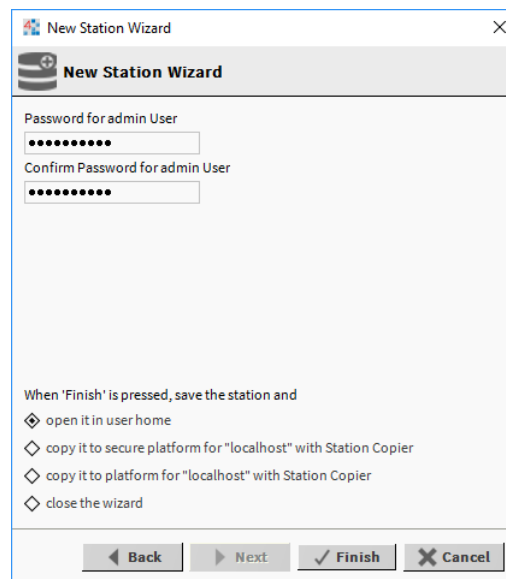


Figure 4-2 New Station Wizard administration

After you exit the **New Station Wizard**, the station is created in the Workbench User Home directory under **stations** and displayed in the Niagara Workbench.

#### 4.1.4 Establishing the platform connection

Commissioning is done in the Niagara Workbench. To do this, establish a connection to the Niagara platform of the Niagara host by choosing **File – Open – Open Platform** from the menu. Select **Platform Connection** for the session type, enter the host IP address and confirm your entries by choosing **OK**.

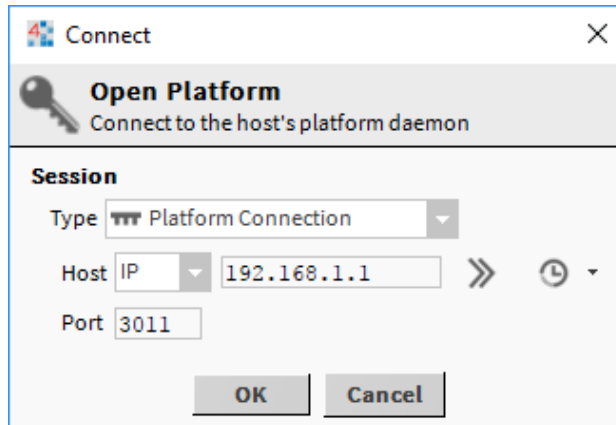


Figure 4-3 Commissioning: establishing a connection

You must now enter the user name and password for the platform connection. The default user name for the ILC 2050 BI is **sysmik** and the password is **intesa**. Use your Windows login credentials for the Niagara Supervisor.

As of Version 4.4, you will be requested to change the passphrase for the controller and the platform access data when first making a platform connection to an ILC 2050 BI.

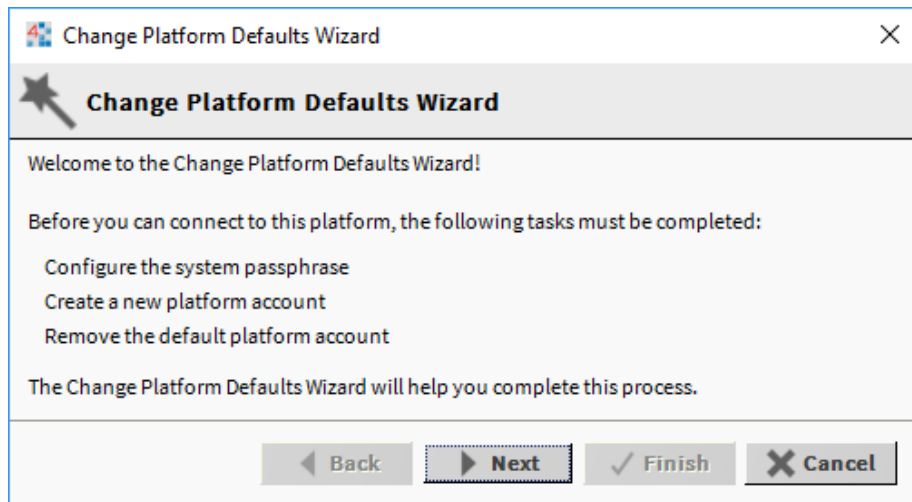


Figure 4-4 Change Platform Defaults Wizard

## Niagara commissioning

This system passphrase is used to encrypt sensitive data such as the password hashes of station users. Enter an individual and secret passphrase that replaces the default passphrase, **intesa**. The passphrase is subject to the same requirements as a platform password: It must be at least 10 characters long and contain at least one number, an upper-case letter, and a lower-case letter.



### NOTE

This system passphrase is important and should not be forgotten!

You also require the current passphrase in order to change the passphrase later. If you forget the passphrase, it can only be reset to **intesa** by installing a clean distribution.

The screenshot shows a dialog box titled "Change Platform Defaults Wizard" with a close button (X) in the top right corner. Below the title bar, there is a star icon and the text "Change Platform Defaults Wizard" and "Configure the System Passphrase". The main content area contains the instruction "Create the passphrase used to encrypt sensitive information on the platform's filesystem:". Below this are two input fields: "New Passphrase" and "Confirm New Passphrase". At the bottom, there are four buttons: "Back" (disabled), "Next" (active), "Finish" (disabled), and "Cancel" (disabled).

Figure 4-5 Configuring a passphrase

Now define an individual and secret platform user with user name and password that replaces the default platform user, **sysmik**.



### NOTE

The platform access data is important and must not be forgotten!

If you forget the access data, reset to user name **sysmik** and password **intesa** using the SCA system shell.

The screenshot shows a dialog box titled "Change Platform Defaults Wizard" with a close button (X) in the top right corner. Below the title bar, there is a star icon and the text "Change Platform Defaults Wizard" and "Create a Platform Account". The main content area contains the instruction "Create a new platform account to use on this platform:". Below this are four input fields: "New Username" (containing "admin"), "New Password" (masked with dots), "Confirm Password" (masked with dots), and "Comment (optional)". At the bottom, there are four buttons: "Back" (disabled), "Next" (active), "Finish" (disabled), and "Cancel" (disabled).

Figure 4-6 Configuring platform access data

All the configured changes are displayed again.

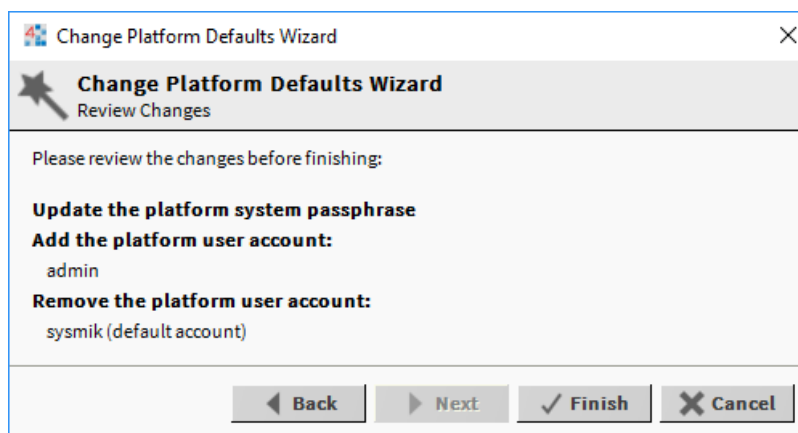


Figure 4-7 Summary of the changes

Confirm with finish to execute the changes and use the platform connection.

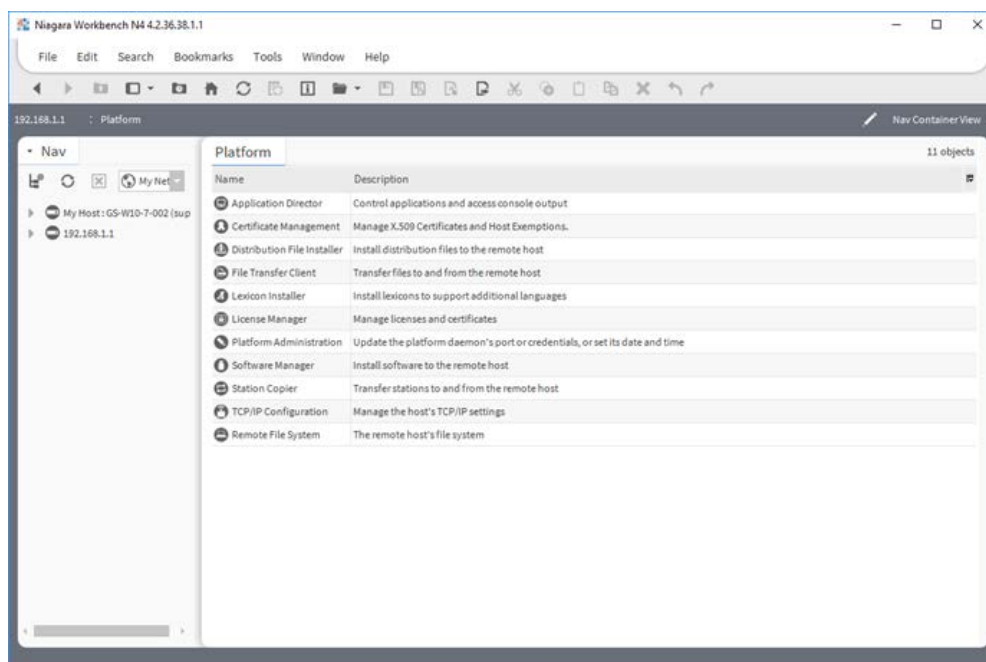


Figure 4-8 Commissioning platform tools

The list of platform tools is now displayed in the workspace of the Niagara Workbench.

## 4.2 Configuring commissioning steps

Commissioning involves many steps, all of which can be performed individually with the appropriate platform tools as needed.

The Commissioning Wizard, which you open by choosing **Platform Administration - Commissioning**, combines all of these steps into one operation.

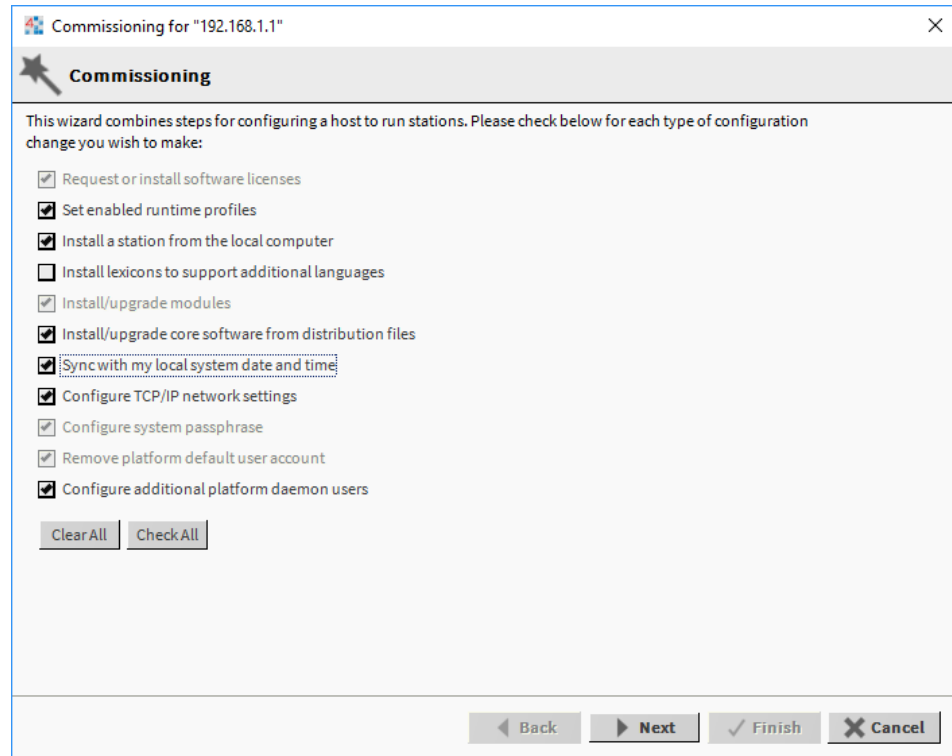


Figure 4-9 Commissioning checklist

Some steps are mandatory. You can manually enable or disable other steps. You subsequently configure each enabled with the help of on-screen prompts. You can interrupt the wizard at any time. The steps are not executed as a unit until all steps have been completely configured.

## 4.2.1 Licensing

All Niagara hosts have a host ID on which licensing is based. Licenses are created, transported and stored as license files. Each license file contains, in addition to the host ID, the Niagara version to which it applies, as well as a number of features that it enables. There are three possible license sources:

License files can be obtained via the sales channel and downloaded from the file system (offline).

All licenses are generated on a publicly accessible license server and can be accessed online by the Niagara Workbench at any time. Licenses can be updated on the license server by purchasing additional options (new features) or as part of software maintenance (higher Niagara version).

The Niagara Workbench simultaneously maintains its own local workbench license database in which all licenses installed by the Workbench are automatically entered. You can maintain this workbench license database in the **Workbench License Manager (Tools – Local License Database menu)**. Many licenses can be synchronized with the license server online so that they can later be installed offline in controllers via commissioning.

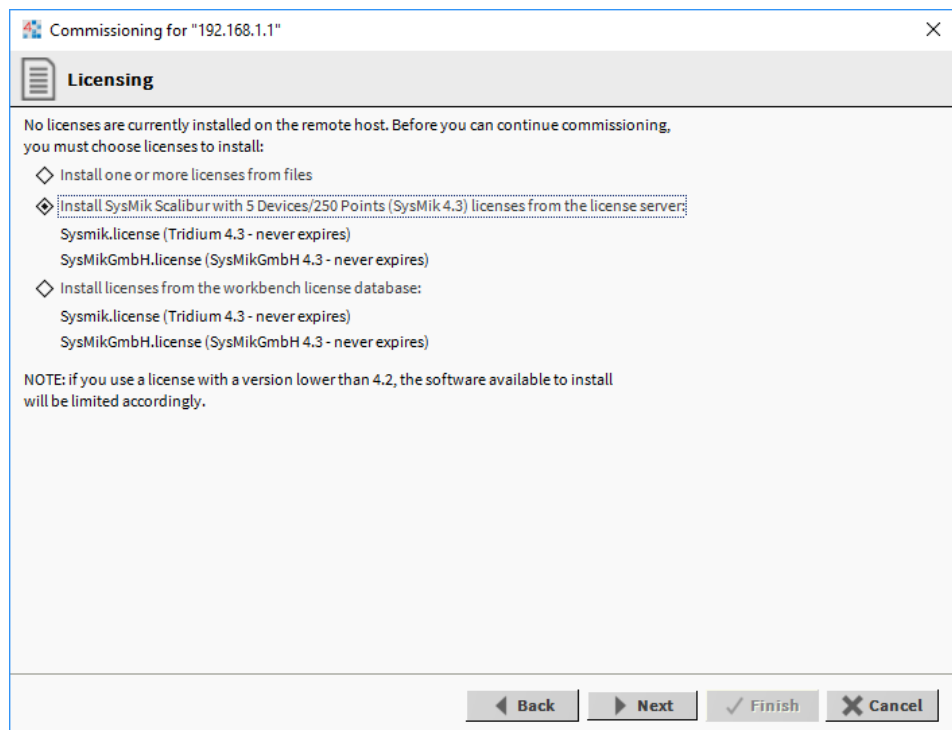


Figure 4-10 Workbench License Manager

In this case, commissioning is done in Niagara Workbench 4.2. The Niagara version of the Niagara Workbench is installed on the Niagara host. As part of ongoing software maintenance, the license on the license server has already been automatically upgraded to version 4.3 with the release of Niagara 4.3. All older Niagara 4 versions can be run without restriction with the license for Niagara 4.3.

## Niagara commissioning

### 4.2.2 Runtime profiles

Niagara is modular. Functions are grouped into modules. Modularization is based on two criteria: the intended application and the runtime profile.

For example, for the intended application of trend data collection, there are three modules for different runtime profiles: history-rt.jar, history-ux.jar and history-wb.jar

The Runtime profile selected during commissioning determines which of these modules should be installed:

The **RUNTIME** runtime profile contains the core functions without a user interface and is always enabled. The module for history collection is history-rt.jar.

**UX** and **WB** must also be enabled if you want the station to have a web visualization. These are user interfaces such as managers or plug-ins. For trend data collection, these are the modules history-ux.jar and history-wb.jar.

The **DOC** option, which contains the online documentation, is less common. Trend data collection is documented in the docUser-doc.jar online user manual

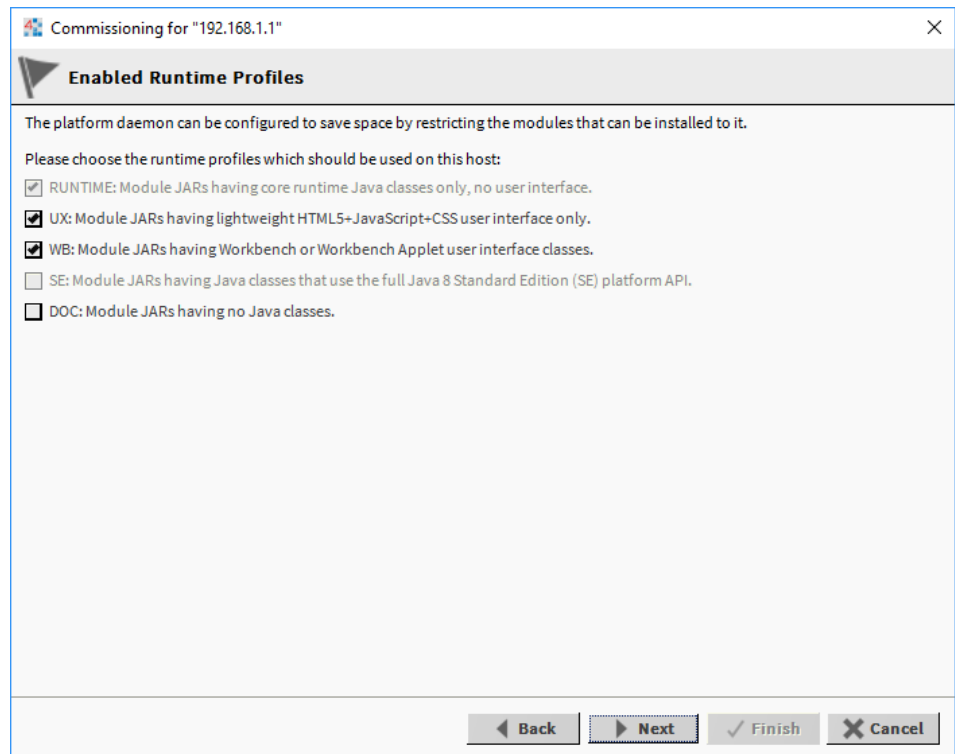
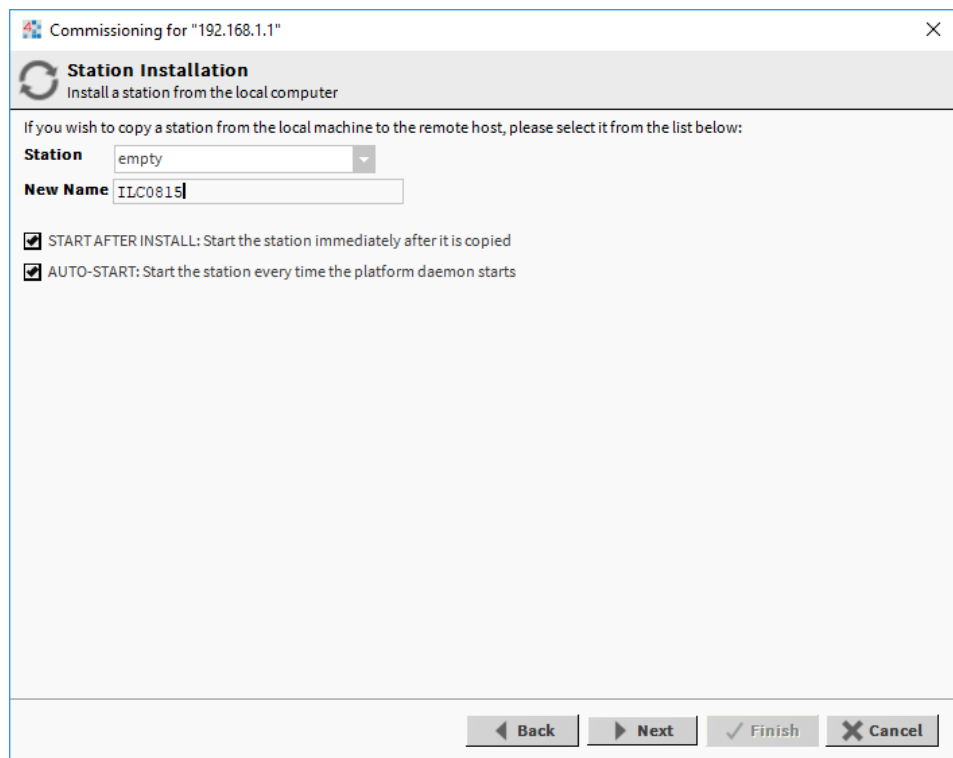


Figure 4-11 Runtime profile

### 4.2.3 Installing a station

If a Niagara Station already exists in the file system of the Niagara Workbench, it can be installed in the host with this commissioning step.



Commissioning for "192.168.1.1"

#### Station Installation

Install a station from the local computer

If you wish to copy a station from the local machine to the remote host, please select it from the list below:

**Station** empty

**New Name** ILC0815

START AFTER INSTALL: Start the station immediately after it is copied

AUTO-START: Start the station every time the platform daemon starts

◀ Back   ▶ Next   ✓ Finish   ✕ Cancel

Figure 4-12 Station installation

## Niagara commissioning

### 4.2.4 Installing modules

In this commissioning step, all modules required to operate the Niagara host are automatically installed. These include a number of basic modules as well as modules that are required by the selected runtime profile (4.2.2) or by the station (4.2.3) to be installed. Module upgrades and other modules can also be installed manually for later use.

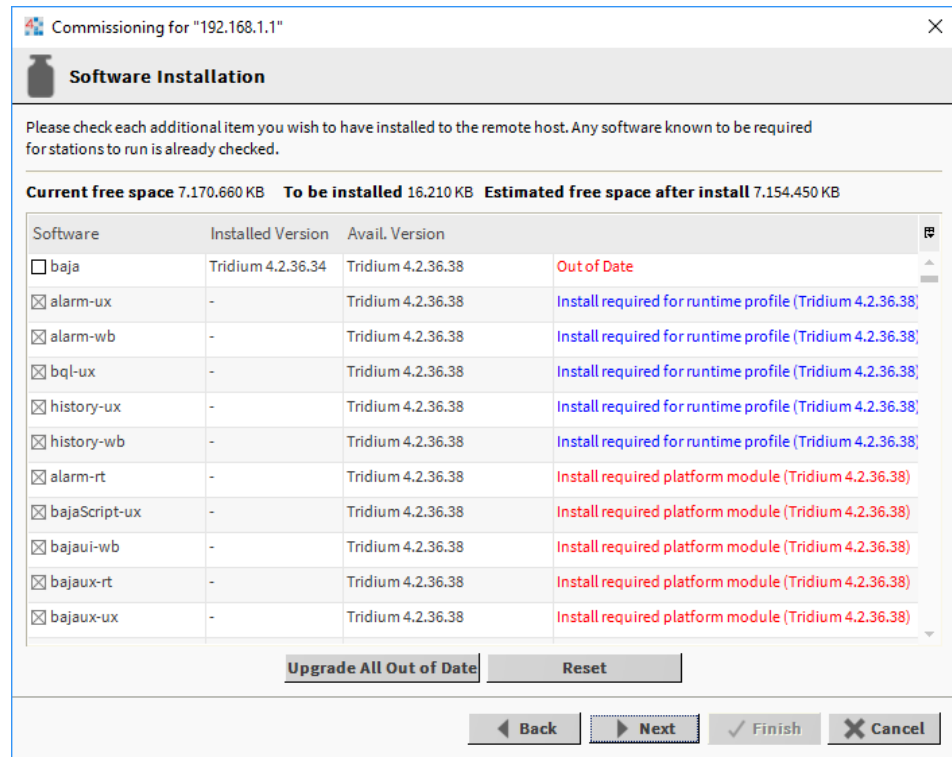


Figure 4-13 Installing software modules

## 4.2.5 Installing the system software

System software includes platform-specific software elements: the operating system, Java Runtime Environment and Niagara Runtime Environment. These software packages are delivered and versioned in the form of distribution files. The distribution files for the ILC 2050 BI are part of the SysMik Niagara AddOns. During commissioning, new versions of the system software are transferred to the Niagara host and installed there.

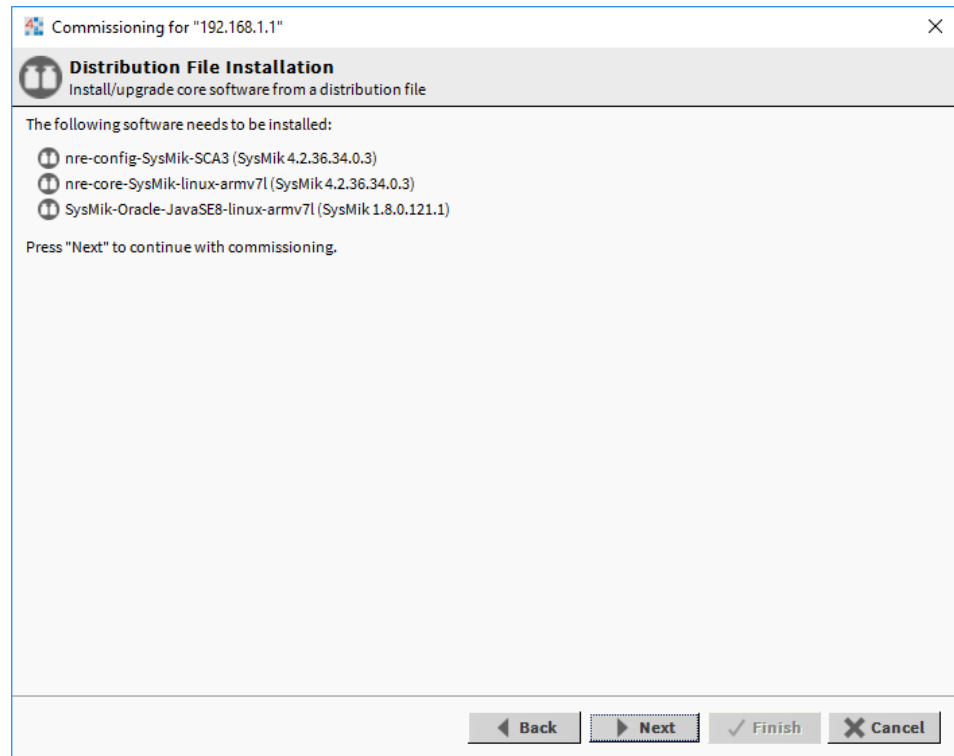


Figure 4-14 Installing the system software

## 4.2.6 Setting the time

During commissioning, the time of day and time zone can also be transferred from the Niagara Workbench to the Niagara host. There is no input screen for this step. If you want to define a different time and time zone than the settings in the Workbench, you can do this outside the Commissioning Wizard using the **Platform Administration** platform tool.

## Niagara commissioning

### 4.2.7 IP settings

In this step, IP settings can be defined for all available network adapters of the Niagara host.

Commissioning for "192.168.1.1"

### TCP/IP configuration

Platform TCP/IP settings

Change the TCP/IP settings of the platform by modifying the values below:

Interface 1 ▲

**ID** eth0

**Description** eth0

**Physical Address** 08:0C:0B:00:07:EA

**Adapter Enabled**  Enabled

**IPv4 Settings** **IPv6 Settings**

**DHCPv4**  Enabled

**DNS Domain**

**IPv4 Address** 10.0.7.42

**IPv4 Gateway** 10.0.0.1

**IPv4 Subnet Mask** 255.0.0.0

**DHCPv4 Server** n/a

**DHCPv4 Lease Granted** n/a

**DHCPv4 Lease Expires** n/a

**DNSv4 Servers**

Undo Changes

Back Next Finish Cancel

Figure 4-15 IP configuration

## 4.2.8 Configuring platform users

The default platform user name and password for the ILC 2050 BI controller are **sysmik** and **intesa** respectively. These are replaced by individual access data when the platform connection is established for the first time. You can create additional platform users during this step.



### NOTE

These platform login credentials are important and must not be forgotten.

If the credentials are lost, they can be reset to user name **sysmik** and password **intesa** using the SCA system shell.

The screenshot shows a window titled "Commissioning for '192.168.1.1'" with a close button (X) in the top right corner. Below the title bar is a header section with a user icon and the text "Platform Daemon Authentication" and "Platform user account information". The main content area contains the instruction: "You may add, remove or edit user accounts the daemon uses to authenticate users by changing the values below:". Below this is a table with the heading "Users". The table has two columns: "Name" and "Comment". There is one row with the name "admin" and an empty comment field. At the bottom of the table area are three buttons: "New User", "Delete User", and "Change Password". At the very bottom of the window are four navigation buttons: "Back", "Next", "Finish", and "Cancel".

Name	Comment
admin	

Figure 4-16 User configuration

### 4.3 Completing commissioning

All information recorded by the Commissioning Wizard is displayed again. You complete the commissioning process by choosing the Finish button. This can take a few minutes. The Niagara host is restarted after the data is loaded.

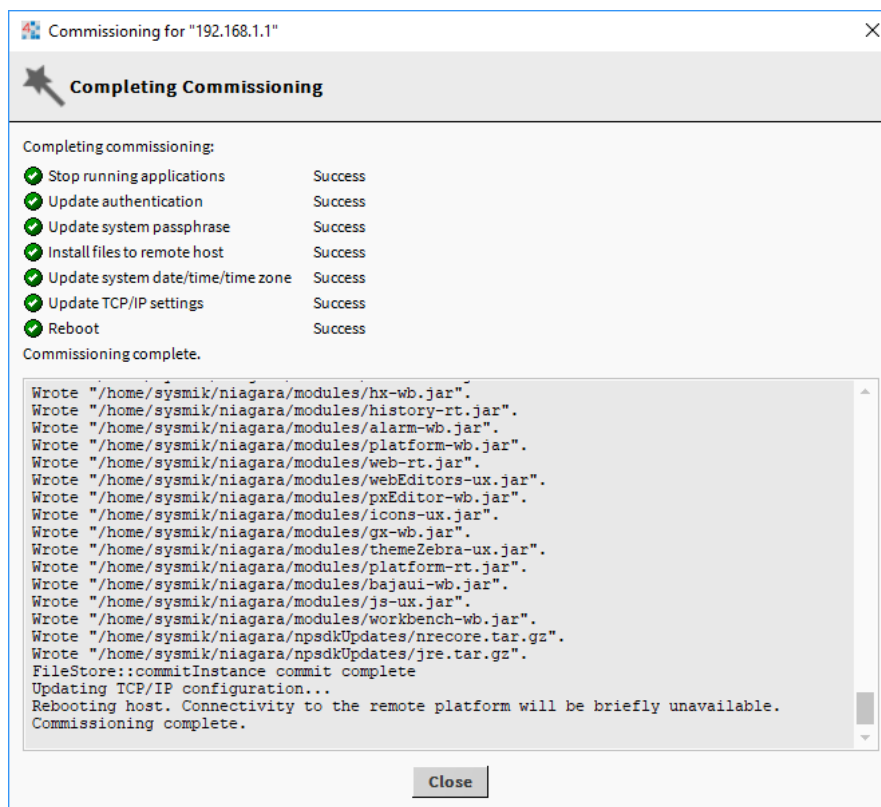


Figure 4-17 Completing commissioning

When the host restarts, any transferred system software is installed, and the Niagara host will restart several times.

The process status LEDs on the ILC 2050 BI indicate the current system status.

**Table 4-1: Process status LEDs (label)**

LED	Meaning
PL	Status of the Niagara platform
ST	Status of the Niagara station
SE	Status of the Sedona Virtual Machine with I/O server

In normal operation, all three LEDs are activated according to the same scheme. It is very easy to see which software processes are currently starting (high processor load) or when a startup process has been completed (low processor load).

**Table 4-2: Process status LEDs (function)**

Process status LED	Meaning
Off	The process has not started.
Flashing yellow with 2 s cycle duration, 10% (on): 90% (off)	The process requires less than 10% of the processor time.
Flashing yellow with 2 s cycle duration, 50% (on): 50% (off)	The process requires between 10 and 50% of the processor time.
Flashing yellow with 2 s cycle duration, 90% (on): 10% (off)	The process requires more than 50% of the processor time.

In addition, during a software upgrade, the process status LEDs indicate the installation process with a chase effect.

Once all three LEDs flash in unison, the commissioning process is finished and the Niagara Station will run on the Niagara host.

## 5 Station engineering

Now that the Niagara host has been commissioned, online engineering of the station can begin. Engineering a station is essentially a type of graphical programming. Required modules are installed and components of different modules are created, configured and connected to each other in the station.

For this activity, it is best to use the Niagara Workbench with the **Nav** side bar for navigating in the station and the **Palette** side bar for selecting components from modules. You can enable both side bars from the **Window – Side Bars** menu.



You can also engineer a station offline in the file system of the Niagara Workbench. However, no application functions are executed in this case.

By loading a station in the Niagara Supervisor, you can edit a station online without a Niagara controller. However, some functions such as I/Os and serial drivers are not available in the Niagara Supervisor. In addition, functions may be restricted by the Niagara Supervisor license.

### 5.1 Components

Components are the building blocks for stations. They are grouped into modules according to application. Components have slots. These can be properties, actions or topics.

An action is an input from which an action is initiated. Actions can be initiated via the local **Actions** menu.

A topic is an output from which an event is reported.

A property can be many different things. Components can be properties of components. This is how the tree structure of the station is created. Properties with less complex data types are used as inputs and outputs, or parameters of the components.

### 5.2 Views

Components have views that show specific aspects of the component. There are various default views that, with a few exceptions, are available for all components.

The **AX Property Sheet** shows visible properties of a component together with their value in a table. Writable properties can be modified here.

The **Wire Sheet** view graphically displays all (lower-level) components of the component as a block diagram. Links between components can be created and deleted in this view.

The **AX Slot Sheet** view shows all (even non-visible) slots of the component in a table. Via the local **Config Flags** menu, you can change flags that control the visibility of slots (**Hidden**), for example.

The **Property Sheet** view is a combination of AX Property Sheet and AX Slot Sheet in HTML5.

The **Relation Sheet** view shows relationships (including connections) of the component to other components.

Each component may also have additional specific views. For example, **UserService** has a specific **User Manager** view that lets you easily create and manage users.

All views of a component can be selected from the local **Views** menu.

### 5.3 Station connection

The Niagara Workbench is used for engineering a station. To do this, establish a new connection between the Niagara workbench of the Niagara host by choosing **File – Open – Open Station** from the menu. Select **Station TLS** for the session type, enter the host IP address and confirm your entries by choosing **OK**.



All Workbench connections are stored in the **Nav** side bar in the nonvolatile memory. If the navigation tree already contains the IP address of the host, you can also create the station connection via the local menu of the **Open Station** IP address without having to enter the IP address again. If the navigation tree already contains a station connection, simply double-click to use the connection.

The public key of the Niagara host is displayed for confirmation upon initial contact with a Niagara host via a secure encrypted protocol (TLS). Afterwards, you log in with user name **admin** and password **Admin12345**, which was assigned when the station was created (4.1.3). If you select the **Remember these credentials** checkbox, the login credentials are stored in the Niagara Workbench and no longer have to be entered in order to establish a connection.



From the **Tools – Manage Credentials** menu you can view, reset and delete all login credentials (without passwords, of course) stored by the Niagara Workbench.

## Station engineering

Once the connection is established, the **Station Summary** view, similar to the navigation tree, displays elements of the station as well as station properties.

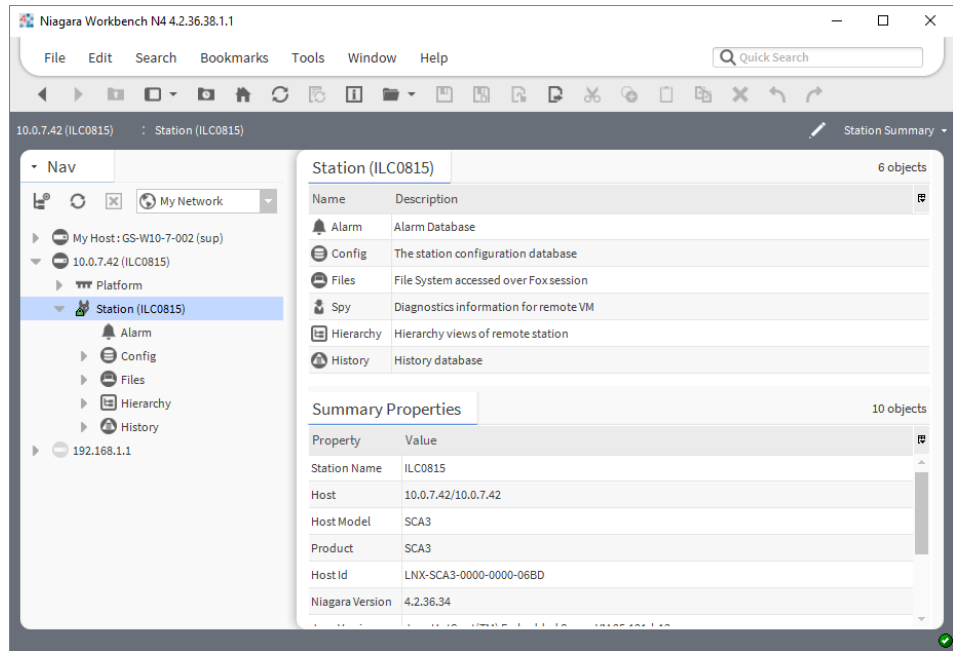


Figure 5-1 Station summary

At this point, we are only interested in the **Config** element. This is the engineering level of the station. All lower-level elements were saved as XML files when the new station was created (4.1.3) and now "live" in the Java Virtual Machine of the Niagara host after the Niagara Station has been installed in the Niagara host. An example of this is the **FoxService** under **Config / Services**, which implements the Fox protocol for communication between the Niagara Workbench and Niagara station.

## 5.4 Hello World – first application

The first exercise is to add two numbers.

### 5.4.1 Installing the module

The **kitControl** module contains an add component. The module includes numerous primitive components that perform arithmetic or logical functions.

Before you can create components of a module in the station, you must install the corresponding module in the Niagara host. You do this with the **Software Manager** platform tool.

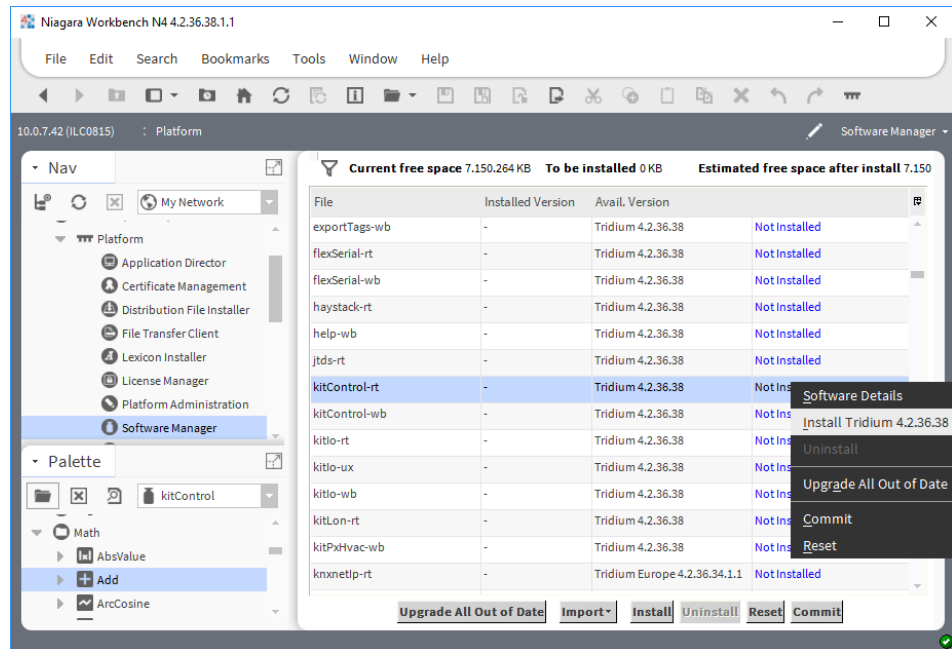


Figure 5-2 Software Manager workbench

The Software Manager displays all modules of the Niagara host (**Installed Version**) and the Niagara Workbench (**Avail. Version**). You can select **kitControl-rt** for installation via the local menu. Choose **Commit** to transfer the module to the Niagara host.

### 5.4.2 Creating the component

Use the **New – Folder** local menu in **Config** to create a new folder called **MyAdd**, in which the application will be implemented. Double-click the newly created component to open the **Wire Sheet** default view.

The **Palette** side bar contains controls. When you move the cursor over these elements, an explanation is displayed at the bottom left of the Niagara Workbench. Open a list of all modules of the Niagara Supervisor installation using the **Open Palette** control. Enter the first few letters of **kitControl** to quickly narrow the selection and then select **kitControl**. The components of the **kitControl** module are now displayed by category in the **Palette** side bar.

Create an additional component in the station by dragging the **Add** component from the **Math** folder in the palette to the **Wire Sheet** of the **MyAdd** folder in the workspace.

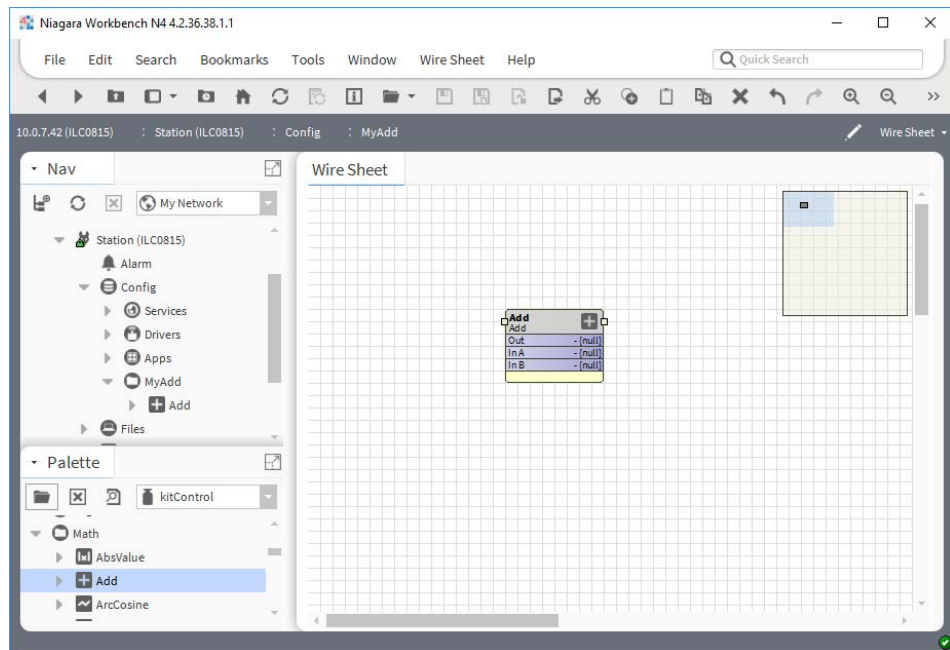


Figure 5-3 Workbench workspace



The content of the workspace can be described using an object resolution descriptor (ORD). You can easily switch between the most recent views by choosing **File - Recent Ords** or **File - Back** and **File - Forward** from the menu.

### 5.4.3 Configuring the component

Double-click **Add** to open the default view for this component, **AX Property Sheet**. All properties of the component and their values are displayed in this view. Enter values for **InA** and **InB** and save your changes to the parameters by choosing **Save**. The **Out** property immediately displays the result of the add operation. The result is displayed in the Niagara Workbench. However, the Niagara station performs the calculation on the Niagara host.

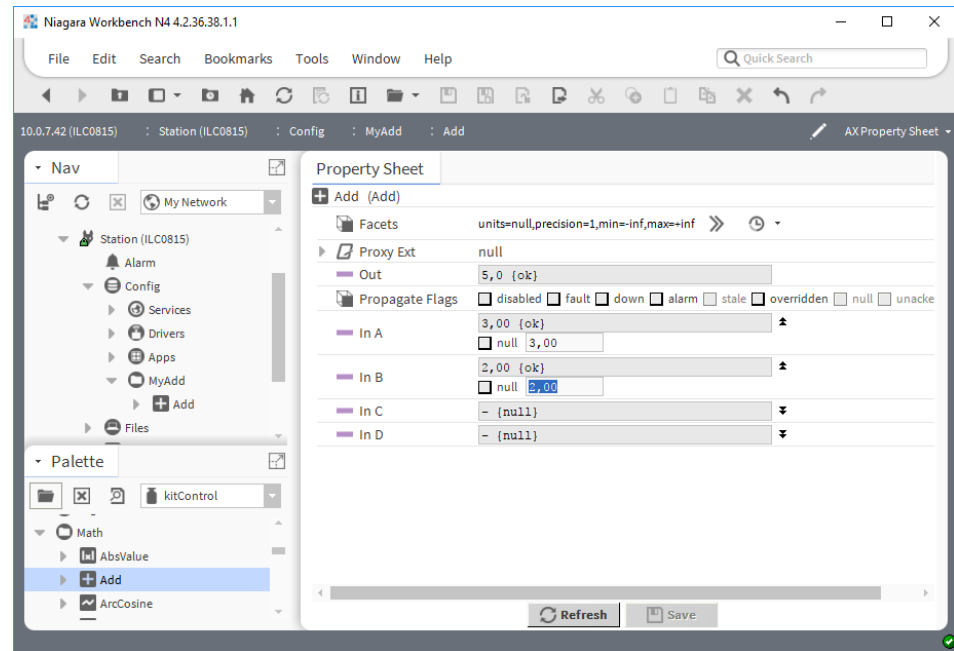


Figure 5-4 Property Sheet

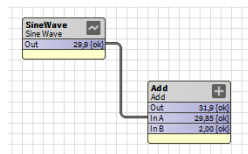
#### 5.4.4 Connecting components - links

You now want to add a component to process a discrete signal (that is, a signal that changes its value at certain intervals) instead of a fixed parameter value.

The kitControl module also contains signal generators. Insert a **SineWave** sine signal generator from the **Util** folder of the **kitControl** palette into **MyAdd**. With the default configuration, this component generates a sine signal with an amplitude of 50.0, an offset of 50.0 and a period of 30 seconds. The output value is calculated every second. This means that the output values fluctuate between 0.0 and 100.0.

To connect the output of the signal generator to an input of the add component, move your cursor over the right third of the **Out** slot of the **SineWave** component in the Wire Sheet of **MyAdd**. The cursor will turn into an arrow ( $\Rightarrow$ ). Now press the left mouse button and drag the cursor (now a closed fist) over the left third of the **InA** slot of the **Add** component ( $\Rightarrow$ ) and let go of the mouse button.

Figure 5-5 Connecting components



A connection between **SineWave.Out** and **Add.InA** is established and displayed in **Wire Sheet**. You can now clearly see how the add component recalculates the output each time the input value is changed.



You can also interconnect components that are displayed in different structures and therefore cannot be displayed on a **Wire Sheet**. To do this, first select a component from the local menu by choosing **Link Mark** for a connection and select **Link From** or **Link To** in the local menu of another component, depending on the desired direction of the connection. This opens a dialog window in which you can select the slots of both components to be connected.

The **Relation Sheet** view of a component displays a table of all connections of the component. These connections can also be deleted here via the local menu.

## 5.5 Points

Niagara uses points for the purpose of normalizing information. Points are the basis for all communication drivers as well as for many other functions such as trend data collection, alarm management and scheduling.

There are 4 types of points: Numeric (floating-point number), Enum (lists), Boolean (true/false) and string (text). A distinction is also made between writable points and read-only points. All point types are contained in the **control** palette.

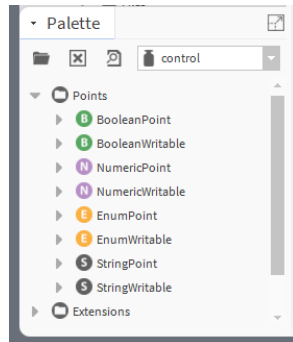


Figure 5-6 Point palette

### 5.5.1 Point value and status

Each point has an **Out** output, which also contains a status in addition to the point value of the respective data type. Important general statuses are:

**{ok}** – everything OK

**{null}** – the point does not contain a value; important for priority control

**{overridden}** – the point is manually overridden

**{alarm}** – the point meets an alarm condition

**{unacked alarm}** – the alarm is still unacknowledged.

### 5.5.2 Writable points

Writable points contain a prioritization mechanism: There are 16 inputs corresponding to 16 priorities. The **Out** point value is defined by the highest-priority input with a signal that is not equal to {null}. Without an input signal, the output goes to **Fallback**. This allows a writable point to be accessed from multiple sources - e.g. heater control via timer, presence signal, manual override, frost protection and fire indicator panel.

Writable points have actions that allow the points to be manually overridden. These actions are also available later in the web visualization. The **Fallback** can be set by choosing **Actions – Set** in the local menu. The **Override** and **Auto** actions can be used to control or release the **In8** input, whereas **Emergency Override** and **Emergency Auto** are used to set **In1** to the highest priority.

## Station engineering

Create a writable point in the **MyAdd** by selecting **New – NumericWritable** from the local menu. Double-click to open the **AX Property Sheet**. Change the input values and notice how priority control works. Connect the **Out** output of the point to **InB** of the add component.

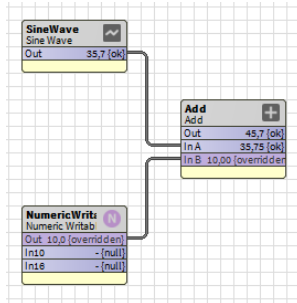


Figure 5-7 Writable points

Overwrite the point value by choosing **Emergency Overwrite** from the local menu.



Slots of components have a **Summary** flag that controls their visibility in the Wire Sheet. Slots without this flag are only displayed if they are connected to another slot. These flags can be changed via the **Pin Slots** local menu.

### 5.5.3 Read-only points

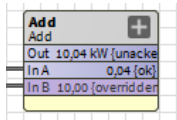
Read-only points have an **Out** output whose value is determined by an internal function and typically cannot be overridden directly by inputs or actions. Many components with processing functions are special derivatives of read-only points. For example, **SineWave** and **Add** from the **kitControl** module are ready-only **Numeric Points**.

Ready-only points from the **control** module have no function and are therefore of no significance for station engineering.

### 5.5.4 Facets

Facets are used to control how points are represented on the user interface. In the **AX Property Sheet** of **Add**, click the **»** edit icon in the line for the **Facets** property. Set the **precision** facet to 2 to display the **Out** output value with two decimal places. In the **units** facet, set the unit to **power / kilowatt** to display the **Out** output value with the physical unit **kW**.

Figure 5-8 Property facet



In the **AX Property Sheet** of a point, you can simply use the **🕒** icon of the **Facets** property to assign one of your recently used facet definitions instead of always having to assign the same facets one at a time for multiple points.

## 5.6 Trend data collection - history

Trend data collection is a standard function of the Niagara Framework. This data is always collected with relation to points. Collection can take place at regular intervals or it can be event-driven when changes are made (COV = change of value). The depth of trend data is configurable. Trend data can be displayed as part of the web visualization. Typically, trend data may also be transmitted to a higher-level Niagara supervisor for more extensive analyses / visualizations or archived in databases via database drivers of the Niagara supervisor.

Trend data is collected by components of the **history** module. These components are not connected to the points via links. Instead, they are placed below the point as **Point Extensions**. In other words, the history extension becomes a slot of the point.

Open the **history** palette. Double-click the **NumericInterval** component in the **Extensions** folder of the **history** palette to display **AX Property Sheet** of the palette component. You can now configure component parameters before they are applied to the station.



### NOTE

Parameter changes in the palette are not saved. After the Niagara Workbench is restarted, all parameters in the palette return to their default values.

Trend data is saved under a name that is defined by the **History Name** property of the history extension. The default value **%parent.name%** is a reference to the **name** slot of the higher-level component.

The capacity of a record can be set via **Capacity**. The default value of 500 in combination with the default **Interval** of 15 minutes ensures that history is recorded over 5 days. Niagara controllers have limited resources and are not suitable for long-term archiving.

Set an **Interval** of one second for real-time trend data collection.

## Station engineering

Extend the station tree in the **Nav** side bar so that you can see the components in **MyAdd**. Using drag and drop, insert one **NumericInterval** component configured in the palette into each of the **Add** and **SineWave** components in the **Nav** side bar.

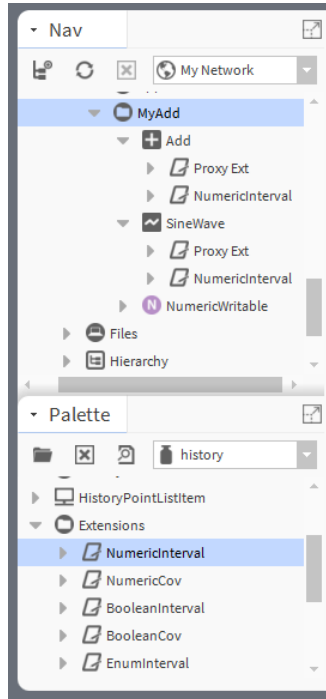


Figure 5-9 Navigation bar



The concept of point extension makes it very easy to copy a configured extension of a point (**Copy** or **Paste** local menu) to other points from the clipboard.

You must now start trend data collection. For individual points, you can do this via the **Enabled** property of the history Extension. The default view of the **HistoryService** under **Config / Services** is better suited for synchronized control of numerous trend data records. In the table of all history extensions, select both lines and start trend data collection by choosing **Enable Collection** from the local menu.

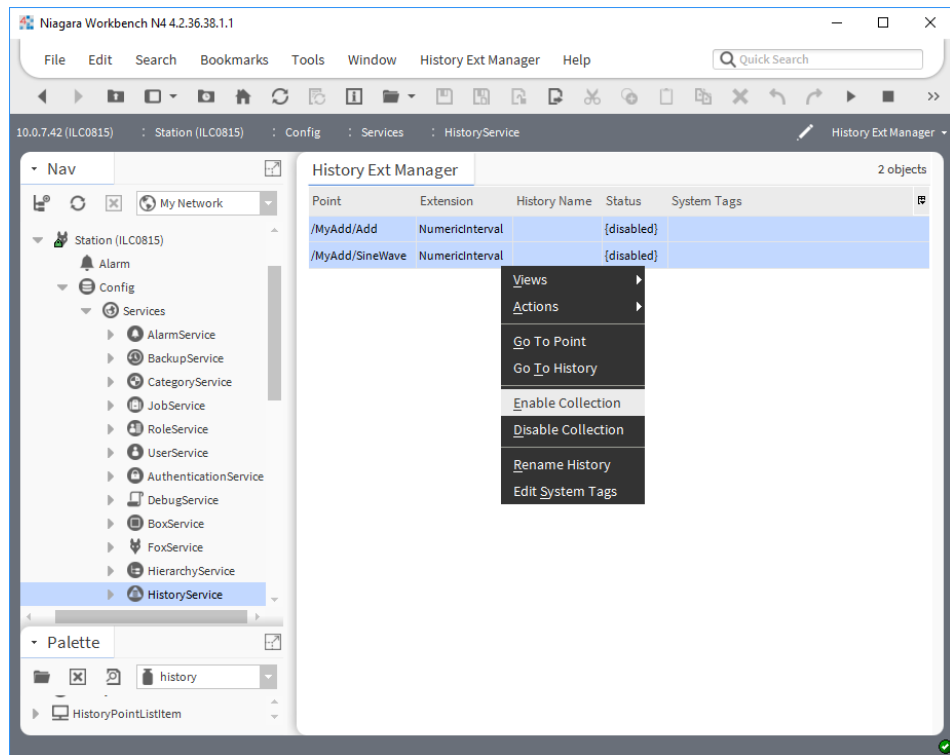


Figure 5-10 History extension manager

## Station engineering

You can view the trend data via **Station / History**. Double-click **Station / History / ILC0815 / Add** in the **Nav** side bar to display the **Add** trend data of the **Add** point in the **Chart** default view. Drag **Station / History / ILC0815 / SineWave** from the **Nav** side bar to the workspace to display both trend data in one diagram.

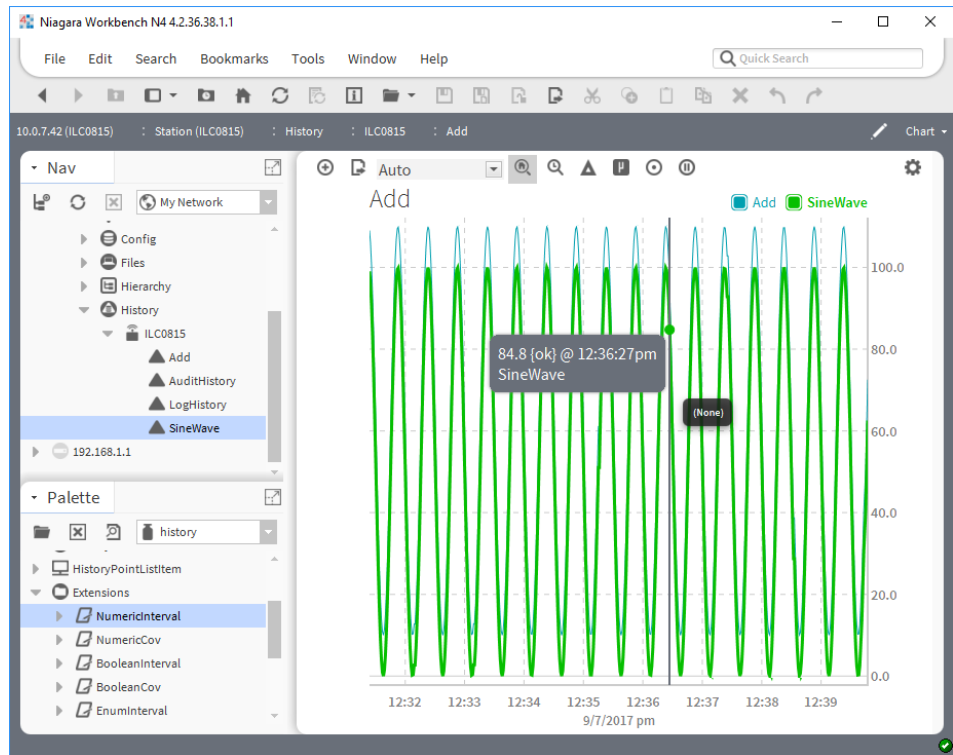


Figure 5-11 Trend data collection



### NOTE

Trend data is saved in the non-volatile memory. Niagara controllers use flash memory for storage, which can only handle a limited number of write cycles. Continuous collection at frequent intervals can damage the flash memory.

Stop history collection via the **HistoryService**.

## 5.7 Alarm management

The Niagara Framework features comprehensive alarm management. Alarm sources generate alarms and route them to alarm classes. Alarms can be routed from alarm classes to different alarm recipients. Typical alarm sources are points that contain an alarm extension as well network and device components of drivers. Typical alarm recipients are alarm consoles as part of a web visualization, e-mails, text messages or printers. Alarms are routed to higher-level Niagara Supervisor stations so that they can be centrally processed. Alarm classes have built-in escalation management. This means that pending alarms can be routed to different recipients with specific delays.

Alarms are routed via the **AlarmService** under **Config / Services**. This service already contains the **DefaultAlarmClass** to which all alarm sources send their alarms. Create a new **MyAlarmClass** by dragging the **AlarmClass** from the **alarm** palette to the **AlarmService**. For the alarm recipient, insert **ConsoleRecipient** from the **Recipients** folder of the **alarm** palette into the **AlarmService**. In the **Wire Sheet** of the **AlarmService**, connect the **Alarm** topic from **MyAlarmClass** to the **Route Alarm** action of the **ConsoleRecipient**.

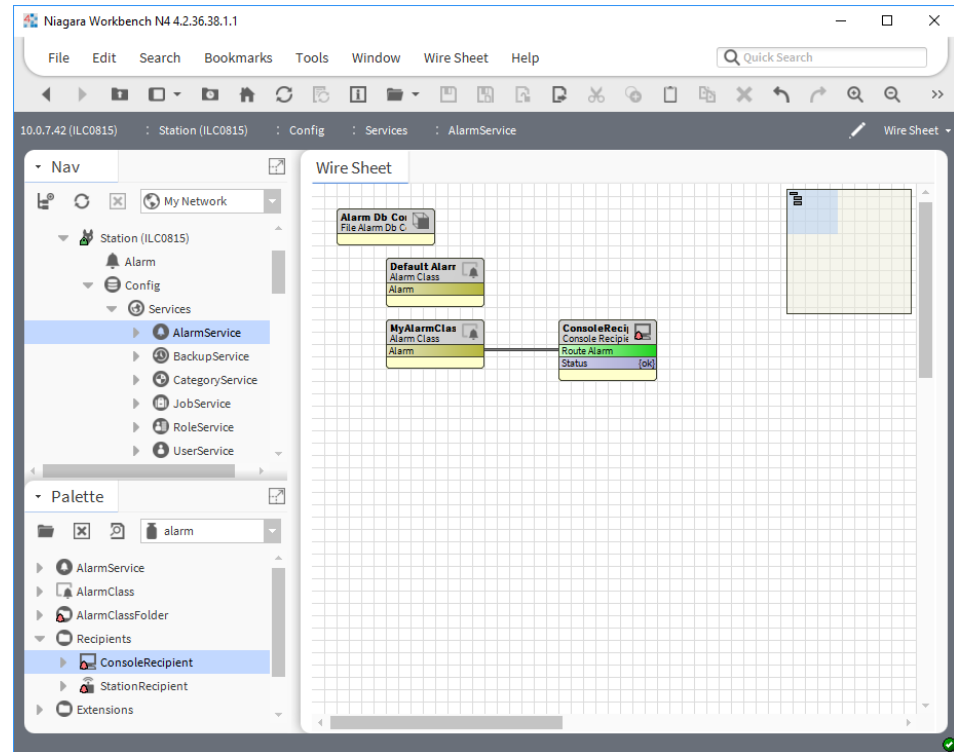


Figure 5-12 Wire Sheet - alarm console

## Station engineering

Extend the station tree in the **Nav** side bar so that you can see the components in **MyAdd**. Using drag and drop, insert an **OutOfRangeAlarmExt** component from the **Extensions** folder of the **alarm** palette in **Add** into the **Nav** side bar. This alarm extension can monitor the value of a numeric point to ensure that it does not exceed limits. An alarm is generated if limits are exceeded.

You can double-click the inserted alarm extension to configure it:

- Select **MyAlarmClass** for the **Alarm Class** property.
- Set **Offnormal Algorithm / HighLimit** to 80 as the upper alarm limit.
- Set **Offnormal Algorithm / LowLimit** to 20 as the lower alarm limit.
- Enter the following alarm text: **Offnormal Algorithm / High Limit Text** = "too high" and **Offnormal Algorithm / Low Limit Text** = "too low"
- Enable both alarm limits in **Offnormal Algorithm / Limit Enable**.

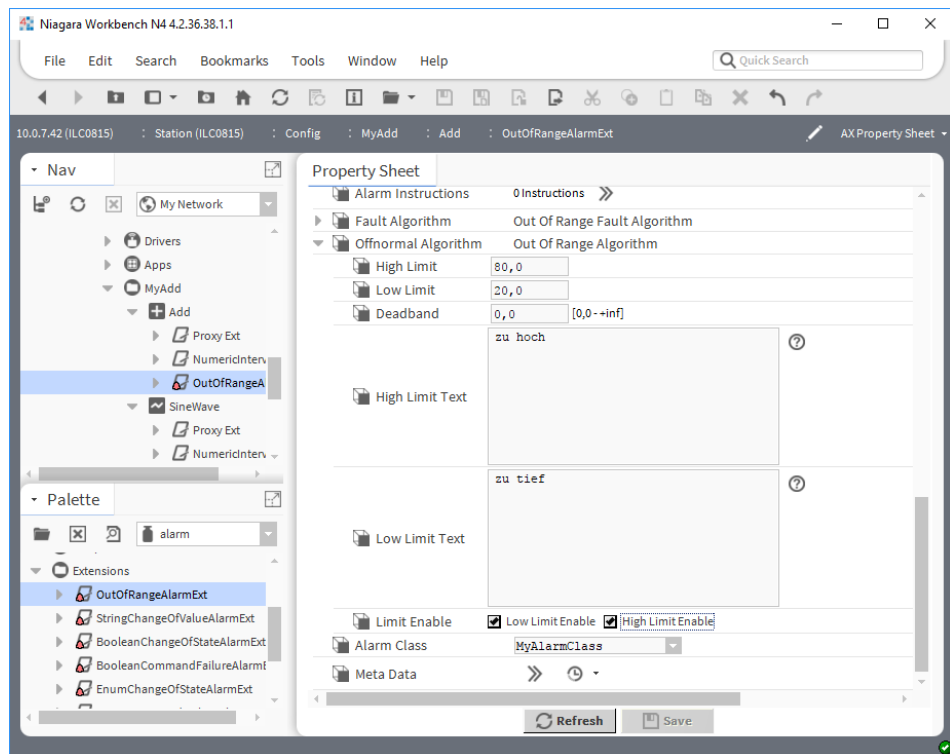


Figure 5-13 Property sheet for OutOfRange alarm

Double-click **ConsoleRecipient** in **Config / Services / Alarm Service** to enable the alarm console, which now displays cyclically recurring alarms. Double-click an alarm to display a table of all alarm events for this alarm source. Double-click an alarm event for a detailed view with a precise record of when and why the alarm was triggered and when the alarm state was exited.

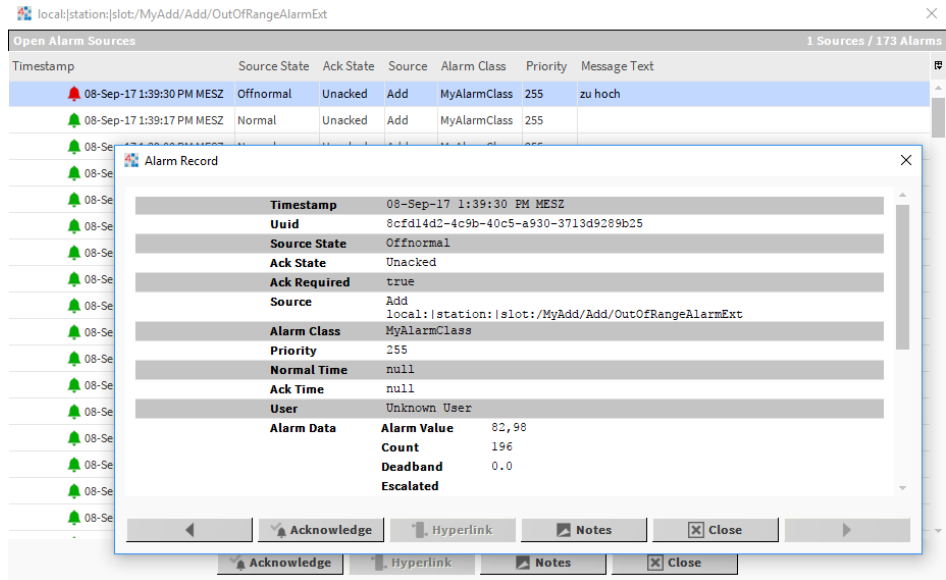


Figure 5-14 Acknowledge alarms

Alarms appear as active alarms (red bell) in the alarm console as soon as the alarm condition monitored by the alarm source is met and the generated alarm has been routed to the alarm console via the Alarm Service. Alarms become inactive (green bell) if the alarm condition is no longer met but the alarm has not yet been acknowledged. Events that require acknowledgement can be specified in the alarm class. Individual or entire groups of alarms can be acknowledged on all levels of the alarm console. A yellow bell indicates acknowledged active alarms. Acknowledged inactive alarms are not displayed in the alarm console.

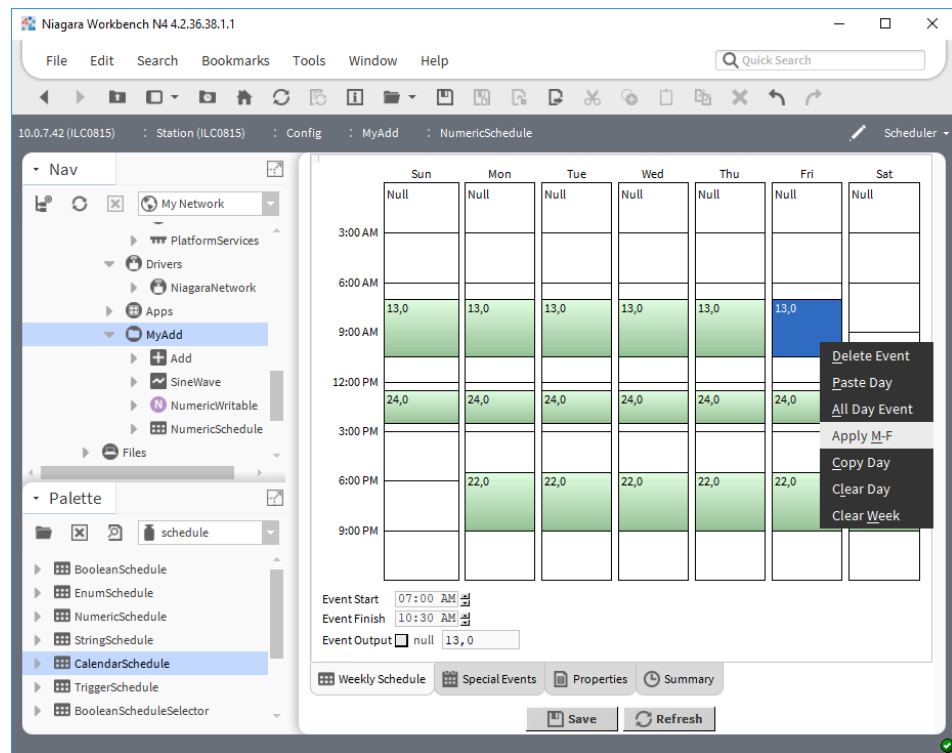
The **capacity** of the alarm database can be specified via **Alarm Db Config** in the **AX Property Sheet** of **Station / Config / Services / Alarm Service**. The alarm database itself can be reached via **Station / Alarms**. The most recent alarms of the station are stored here depending on the configured capacity. The entries can be displayed and deleted here.

## 5.8 Scheduling

Scheduling in the Niagara Framework is handled by components of the **schedule** module. The **CalendarSchedule** component has a Boolean output that is activated on configurable days. **NumericSchedule** has a numeric output that is controlled via configurable daily schedules for each day of the week as well as exceptions.

Insert a **NumericSchedule** component from the **schedule** palette into **MyAdd** and connect the **Out** slot to the **In16** slot of **NumericWritable**. Double-click **NumericSchedule** to open the **Schedule** manager. In a weekly schedule, select a time range by moving the cursor to the start time, pressing the left mouse button, moving the cursor to the end time, and releasing the mouse button. In the **Event Output** field, enter the value to be displayed in the schedule for this time range. Make other entries in the schedule an copy them to other days using the local menu.

Figure 5-15 Calendar / schedule workbench



In the **Special Events** tab, you can configure exceptions (such as public holidays) and special daily schedules for these holidays that deviate from the weekly schedule.

In the **Properties** tab, you can limit the validity period of the schedule and you can define the default output value of the **Default Output** schedule.

## 5.9 Web visualization

The Niagara station has a web server that can be configured via **Station / Config / Services / WebService**. You can connect to this web server via a web browser. To do this, you must sign in to Niagara Station with valid credentials. Depending on your user settings, you can access views and data of the station. This can range from basic visualization pages for monitoring a process to a comprehensive web workbench.



### NOTE

The **WB** and **UX** runtime profiles must be enabled as a prerequisite for web visualization. Only then will the required modules and visual elements be installed in the Niagara platform.

The runtime profiles can be activated during the commissioning process (4.2.2) or via the **Configure Runtime Profiles** button in the **Platform Administration** platform tool.

### 5.9.1 User-defined views

Each component has views. These can be default views like the **Wire Sheet** or component-specific views like driver managers. In addition to these programmed views, which are part of Niagara modules, user-defined views can be configured during station engineering. These user-defined views are comprised of widgets. **Widgets** are graphical elements of the user interface that can interact with the station.

In the local menu of MyAdd, select **Views - New View** to create a new view for this folder. Assign **MyAddView** as the **View Name**. This automatically derives the name of the PX file in which the design of the view is stored. By storing the design in a PX file, the view can be used for multiple components once it has been configured.

#### 5.9.1.1 PxEditor

The **PxEditor** of the new **MyAddView** view in the Niagara Workbench. Here, new widgets can be added to the view or existing widgets can be configured. The view is displayed in the workspace of the PxEditor. The different widgets can be selected, moved or deleted.


The side bars on the right side of the PxEditor help you create the view. These side bars can be enabled or disabled using the  button.

The **Bound Ords** side bar contains a list of all connections of the view to the station via Ords (object resource descriptors).

The **Widget Tree** displays all widgets of the view in a tree structure. This side bar provides easy access to widgets, especially for more complex views with multiple panes and superimposed widgets.

The **Properties** side bar displays the properties of the selected widget and allows you to configure them. Double-click a widget to open the **Properties** window, which has the same function. This window is more effective for engineering if you have a smaller screen.

#### 5.9.1.2 PxViewer

Double-click **Config / MyAdd** to display the new, empty **MyAddView** in **PxViewer** display mode. You can switch between the PxViewer and the PxEditor at any time by choosing **PxEditor / Toggle View/Edit Mode** or **PxViewer / Toggle View/Edit Mode** from the menu or using the corresponding  button.

## Station engineering

---

### 5.9.1.3 Widgets

Widgets are graphical elements of the user interface. Process data of the station can dynamically control (animate) properties of the widgets via configurable bindings. Widgets can also adopt user entries, thereby influencing the data of the station.

### 5.9.1.4 Manually configuring a basic widget

The **Widgets** folder in the **bajai** module contains universal basic widgets of the Niagara Framework. The module also includes geometric primitives in the **Shapes** folder, as well as **Panes**, which control the layout of the view.

#### Adding a widget

Open the **PxEditor**. Drag and drop the **Label** widget from the **Widgets** folder of the **bajai** palette to the workspace in the view.

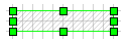


Figure 5-16 Adding a widget

#### Static configuration of a widget

Double-click the widget in PxEditor to display the **properties** of the widget. Enter the character string "MyAdd" for **text**.

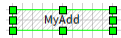





Figure 5-17 Configuring a widget

### Widget animation - creating a binding

The next step is to animate some properties of the widget according to the value of a point of the station. To do this, you must first create a binding to the point. Click the **Add Binding** icon  in the **Properties** window and select **bajauri:ValueBinding**. Click the Browse button  of the **ord** field to enter an object resource locator for a point. You can click the right side of the  icon to use the **Component Chooser** tool to select the **Config / MyAdd / Add** point in the station tree.

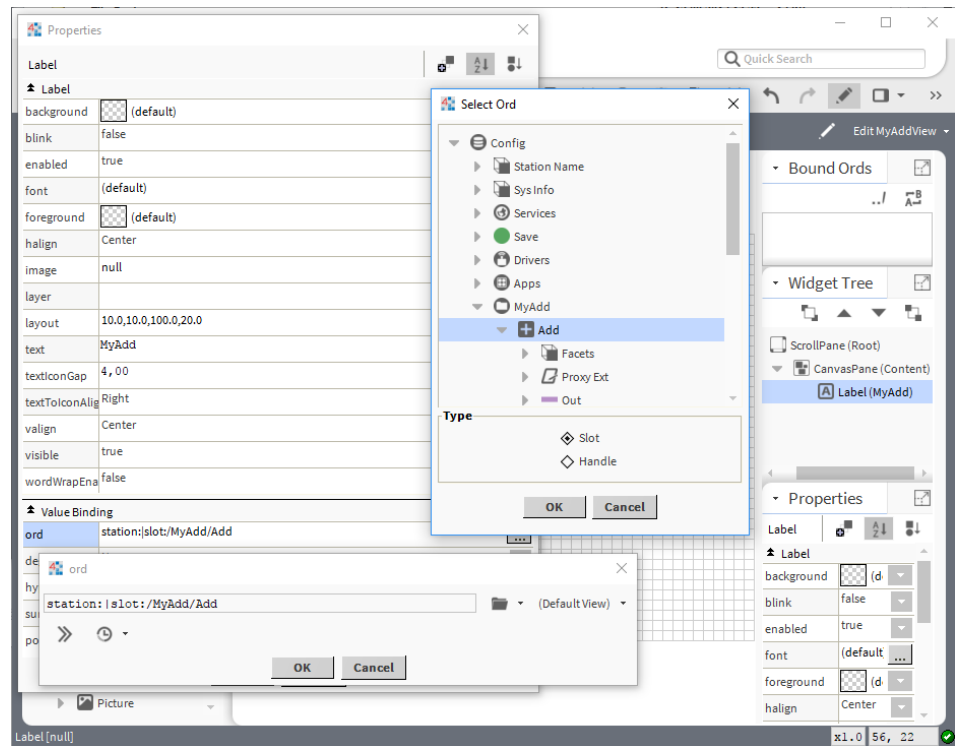


Figure 5-18 Widget binding

## Station engineering

### Animating a widget property

To change the background color according to the point value, select **Animate** from the local menu for **background**. Under **Binding**, the correct (and only) binding of the widget from the numeric point to the simple color value is already selected along with the matching **Converter Type I Numeric To Simple**. Insert several value ranges (Numeric) and define different background colors (Simple).

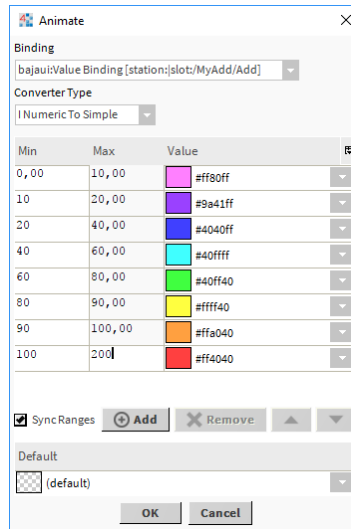


Figure 5-19 Defining widget properties

As a result, the background color changes depending on the value of the add component.

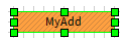


Figure 5-20 Modified widget

All properties of a widget can be animated in this way.

### Animating strings

The **text** property is a static "MyAdd" text with a unique reference to the **Station / MyAdd** folder. If you want to use a widget multiple times, it is best to animate and relativize all references to the station. Animate **text** via the **Animate** local menu by specifying the formatting rule for the text to be displayed. To do this, enter **%parent.name% = %out.value%** as the **BFormat** string in **Object-To-String** converter type.

When the widget is displayed, the BFormat string is converted: All elements enclosed by % signs are replaced by the values of the enclosed properties of the point specified by the binding.

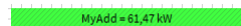



Figure 5-21 Animated BF string

### Relativizing bindings

The **Bound Ords** side bar of the **PxEditor** displays all references to the station that are used in widget bindings. It currently displays absolute Ords with the **station** as the reference point. Click the **Relativize Ords** icon  to convert these absolute Ords into relative Ords (the component of the station to which the view is assigned becomes the reference point).

To see the resulting advantage, copy the **Config / MyAdd** folder in **Nav** using the local **Copy** menu and paste a copy in **Config** using the local **Paste** menu. Since the name **Config / MyAdd** already exists, a new name, **MyAdd1**, is proposed. Use this name. You now have two copies of your addition folder. Both have a user-defined view based on the same px file. Because of the relative Ords, the binding points to the respective lower-level **Add** point and the corresponding folder name is displayed correctly.

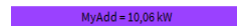


Figure 5-22 Absolute Ords

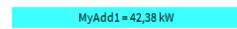




Figure 5-23 Relative Ords

#### 5.9.1.5 Partially configured widgets

The user interface can be fully configured manually, although this is quite complex. Pre-configured widgets are already available for common visualization elements.

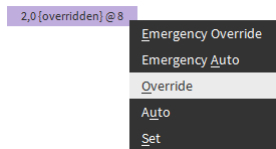
The **kitPx** module contains universal widgets and already includes standard bindings and animations.

From the **kitPx** palette, add a **BoundLabel** widget to the workspace of the **PxEditor**. This widget already contains a binding and an animated **text** property. You only need to specify the Ord of a point here in order to implement a simple value display. Click the Browse button  of the **ord** field to enter an object resource locator for a point. You can click the right side of the  icon to use the **Component Chooser** tool to select the **Config / MyAdd / SineWave** point in the station tree.

### 5.9.1.6 Configuring widgets with wizard assistance

There is an even easier approach: Drag and drop the writable point **Config / MyAdd / NumericWritable** from **Nav** to the workspace of the PxEditor. The Ord is now already defined. The **Make Widget** now appears and displays all available visualization options for selection on the left side. On the right side, the widget resulting from the selected options is displayed with all of its properties. Select **Bound Label** on the left side and activate the **Format Text** and **Status (Color And Blink)** options. Choose **OK** to include the value display in the user-defined view. The status of the point is animated by the background color. Since this is a writable point, the widget automatically provides the actions for manual override via a local menu.

Figure 5-24 Widget with wizard



Drag the writable **Config / MyAdd / NumericWritable** point into the PxEditor workspace. Select **From Palette** in the **Make Widget** and the **SetPointSlider** widget from the **kitPx** palette to insert a value entry for the **Fallback** of the point via a slider into the user-defined view. However, for inherent reasons, this entry does not have an effect on the **out** point value if the point is overridden by the schedule with priority 16 or manually with priority 8.

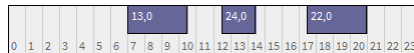
Figure 5-25 Widget wizard - slider



### 5.9.1.7 Visualizing system functions

Drag the **Config / MyAdd / NumericSchedule** schedule into the workspace of the PxEditor and select **Workbench View** and **Current Day Summary** in the **Make Widget** wizard in order to insert the schedule of the current day into the user-defined view.

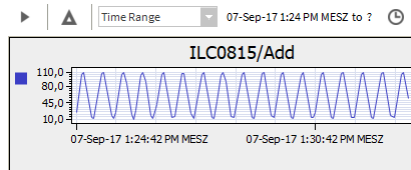
Figure 5-26 Visualized calendar function



The widget has a hyperlink to the default view of the schedule. Of course you can always make changes to the widget configuration. Set the **hyperlinkActive** property to **false** to disable the hyperlink.

Drag the **Config / MyAdd / Add / Numeric Interval** history extension into the workspace of the PxEditor and select **Workbench View** and **History Chart** in the **Make Widget** wizard in order to display the collected trend data in the user-defined view.

Figure 5-27 Widget with history chart visualization



Drag the **Config / Services / AlarmService / ConsoleRecipient** alarm console into the workspace of the PxEditor and select **Workbench View** and **Alarm Console** in the **Make Widget** wizard in order to create the alarm console in the user-defined view.

### 5.9.1.8 Widgets for building automation

The **kitPxHvac**, **kitPxGraphics** and **kitPxN4svg** modules contain specific building automation widgets that allow you to quickly create attractive system visualizations. Important system components such as boilers, fans, pumps, valves, registers, sensors, air ducts and pipelines are assembled from the modules and animated by points of the station.

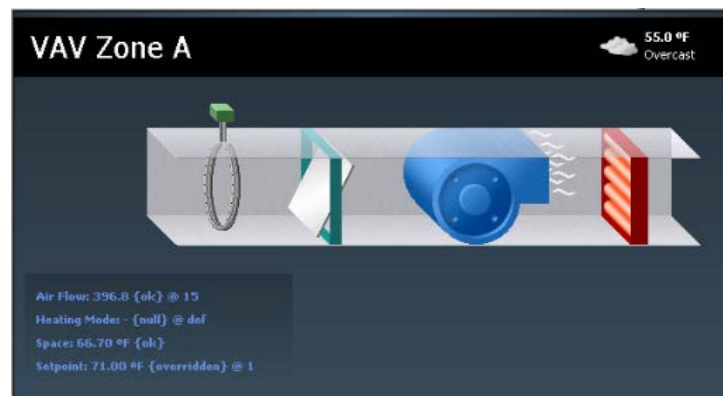


Figure 5-28 HVAC kit graphic (example)

## 5.9.2 User management

A web visualization is closely connected to user management. By configuring a user, you control exactly what the user can view in the web visualization. You must also define corresponding user permissions.



The **admin** user has administrator privileges. This is ideal for engineering because **admin** has access to all elements of the station. For security reasons, this user should not be used for operators of a web visualization. Instead, login credentials with minimal access privileges should be created for each user. This allows recorded user actions to be traced to specific individuals.

User permissions are defined in multiple steps:

### 5.9.2.1 Restricting access to slots

Slots of a component can have an **Operator** flag that can be defined or deleted during engineering in the **AX Slot Sheet** view via the **Config Flags** local menu. For points, typically the **Out** slot carries the operator flag. For writable points, the **set** actions for manual control or **override, auto** actions for manual override also carry the operator flag.

All slots with the operator flag require only operator permissions to read, write and invoke, whereas all other slots have corresponding administrator permissions. The goal is to only assign operator rights to a user of visualization pages and thereby restrict permissions accordingly

### 5.9.2.2 Restricting access to components

Categories can be assigned to components, files and trend data records. Users only receive access permissions for categories that are needed for their task. Typical uses for categories include trades or services.

All components can be individually assigned categories via the **Category Sheet** view. If the **Inherit** checkbox is selected, the assignment of the higher-level component in the component tree is applied. This view also contains a **CategoryService** button that takes you directly to the **Category Browser** default view of the Category Service in **Station / Config / Services / CategoryService**. The Category Browser allows you to efficiently assign all components, files and histories to existing categories.

Two categories, **User** and **Admin**, already exist. All components are assigned to the **User** category. You can define additional categories using the **Category Manager** view in Category Service.

### 5.9.2.3 User roles

Permissions are defined in detail in user roles to facilitate the configuration of user permissions for many users. The user is then simply assigned one or more of the configured user roles.

User roles are defined in **RoleService** under **Station / Config / Services** in the **Role Manager** default view.

In the **Role Manager**, create a new user role using the **New** button and give it the name **Service**. Click the **»** edit icon in the **Permissions** line and in the **Permissions** dialog that appears, enter operator rights for read (**R**), write (**W**) and invoke (**I**) in the **User** category.

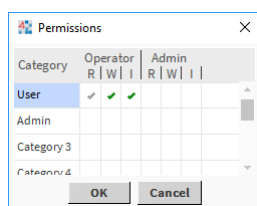


Figure 5-29 User management

### 5.9.2.4 Users

Users can be created and maintained in the **AX User Manager** view of the **Station / Config / Services / UserService** user service. Each user has a user name and password for authentication as well as other user-specific properties.

Create a new user in the **AX User Manager** with the **New** button and assign the name **WebUser** and, under **Authenticator**, password **WebVis12345**. Select the user role **Service** under **Roles**.

## Station engineering

### 5.9.2.5 Navigation

You must now configure the user-defined view for **MyAdd** as the home page for the web server. This is done using a **\*.nav** navigation file. Create a new **nav** folder in **Station / Files** via the local **New Folder** menu. Create a new **nav** file in **Station / Files / nav** via the local **New / NavFile.nav** menu and assign the name **MyAdd.nav**. In the **Nav File Editor** default view of the new nav file, double-click the **Home** line. In the **Edit** dialog that appears, use the **Component Chooser** to enter the Ord of the **MyAdd** Folder in **Target Ord**.

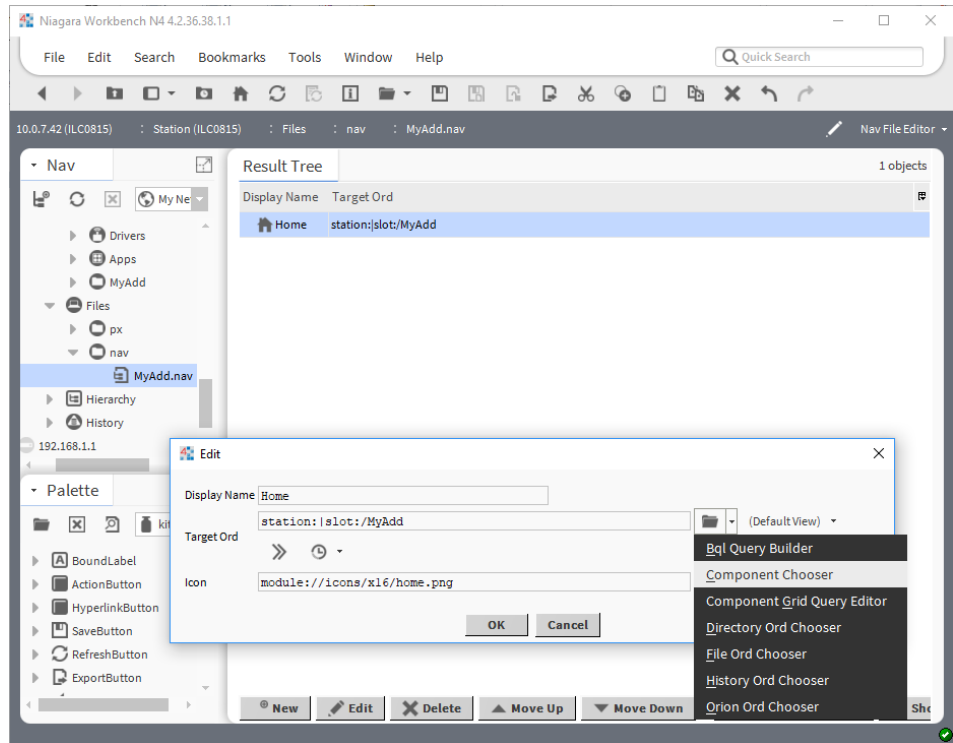


Figure 5-30 Station navigation

In addition to the home page, a basic navigation structure can be added. You must add a **Home** branch to the navigation tree. You can then configure a suitable navigation structure under this branch. To do this, in the **Nav File Editor**, drag the following components from the **Nav** side bar into the **Result Tree** directly below **Home**:

- Station / Config / Services / AlarmService / ConsoleRecipient
- Station / History / ILC0815 / Add
- Station / Config / MyAdd / NumericSchedule

In the **Edit** dialog of the three entries, enter **Alarm**, **Trend data** and **Schedule** in the Display Name field.

In the **User Manager** for the user **WebUser** user under **Nav File**, select the **nav / MyAdd.nav** nav file.

### 5.9.2.6 Web profiles

The Niagara Framework has several defined profiles that determine which functionality is available on the web pages. This functionality ranges from simply display of views from

the nav file to the web workbench. For each user, a profile can be assigned under **Default Web Profile** for contact via the desktop web browser or under **Mobile Web Profile** for contact via browsers of mobile devices. For **Web Users**, use the **User Manager** to set all elements except **Enable Nav Tree Side Bar** and **Enable Nav File Tree** to **No** in the **Default Web Profile** type of **HTML5 Hx Profile**.

### 5.9.2.7 Htpps and certificates

The web server of the Niagara station can be configured via **Station / Config / Services / WebService**. You have a range of options here. Under **Htpps Cert**, you can select the certificate with which the web server will identify itself. Certificates can be created and maintained using the **Certificate Management** platform tool.



#### NOTE

The supplied **tridium** certificate can be used for the time being, but it should be replaced with a trusted certificate if secure communication is required.

When you log in to the web server, the web browser will display warnings because the Tridium certificate is untrusted.

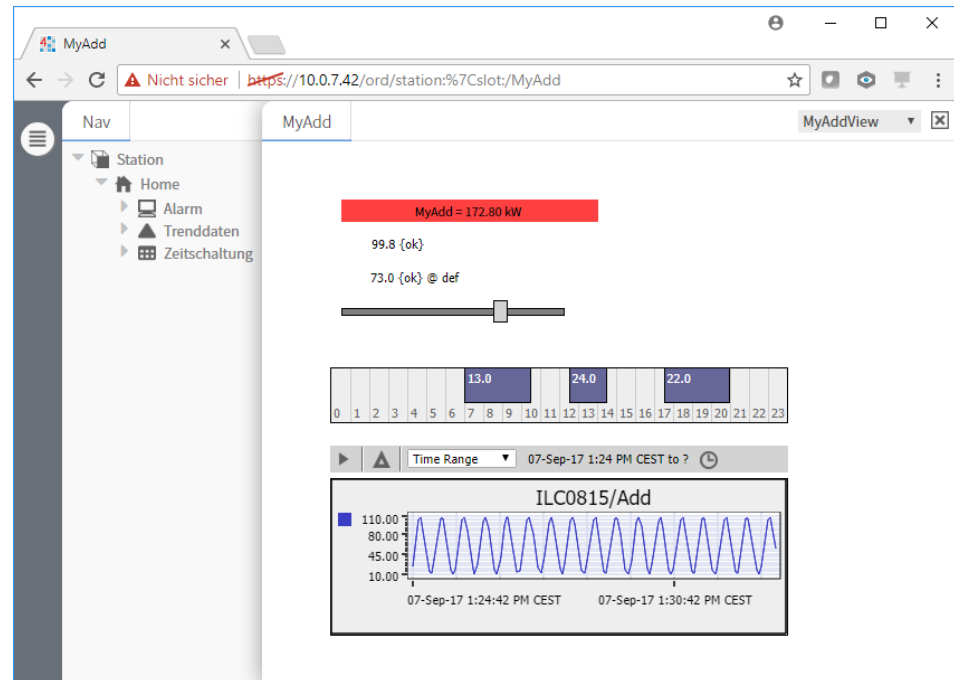


Figure 5-31 Web server warning

## 5.10 Drivers

Drivers play an essential role in the Niagara Framework. They are the interfaces to all higher- or lower-level third-party systems. They implement communication protocols, database query languages or web services and normalize information from third-party systems as Niagara points. The Niagara Driver Framework makes it very easy for Niagara developers to program new drivers for a wide range of applications. The choice of drivers in the Niagara Community is equally large: Over 100 drivers are available, with more and more being added all the time.

Below **Config**, there is a Drivers **folder** containing all communication drivers of the station. Drivers have a uniform structure for the most part. **Devices** are created below the **Network** component with protocol-specific properties. These **devices** have **points**. The user interface, which consists of driver managers, is also normalized. The **Network** component typically has a default **Device Manager**, which helps you create and maintain device components in the Niagara station. The **Points** folder of a device typically has a **Point Manager** as a default view in which points can be created and maintained.

Niagara controllers support serial and IP drivers, whereas the Niagara Supervisor only supports IP drivers. Some drivers can only be used for certain platforms. This includes database drivers that can only be licensed for use with the Niagara Supervisor, or the SysmikScalo driver, which is only used with the ILC 2050 BI for controlling the modular I/O of the controller.



Points of Niagara communication drivers have a driver-specific proxy extension, which forms the link between the communication protocol and the Niagara point. This proxy extension is important from a licensing perspective. Each proxy extension requires a point license. These communicative points are called **proxy points** to distinguish them from other points.

### 5.10.1 I/O driver - SysmikScaloNetwork

The **sysmikScalo** module contains the driver for controlling the modular I/O of the ILC 2050 BI. Open the **sysmikScalo-rt** module via the **Software Manager** platform tool. Additional modules may need to be installed due to existing dependencies.

Double-click **Station / Config / Drivers** to open the **Driver Manager**, where you can easily create drivers in the station. Click the **New** button to display a list of all the drivers included in the Niagara Workbench modules. Select **Sysmik Scalo Network** to create the network.

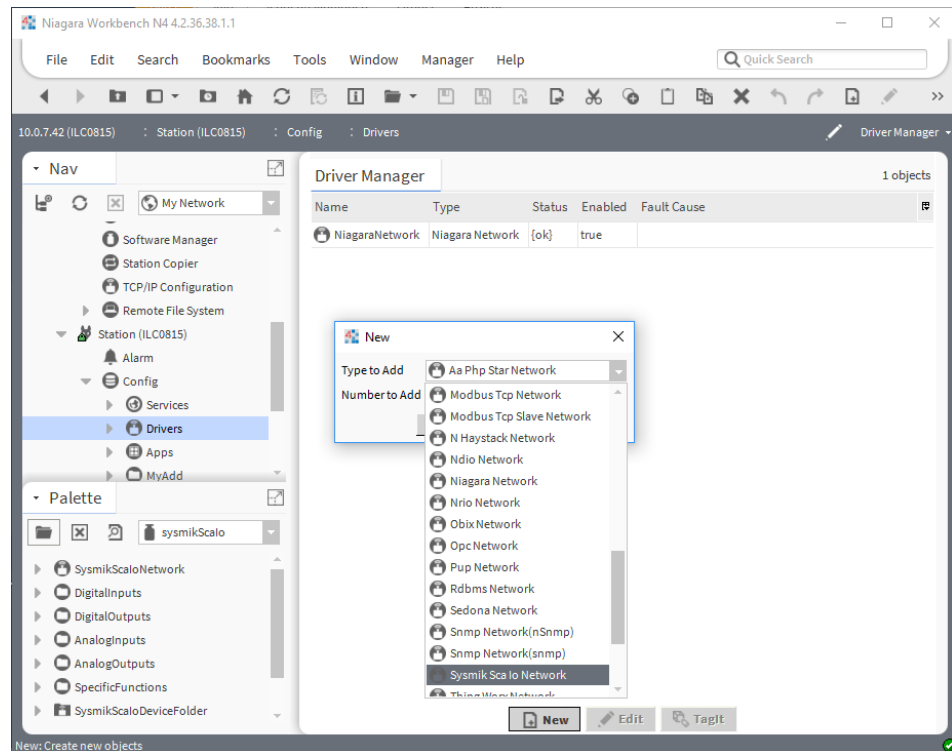


Figure 5-32 Driver management



I/O terminals are controlled largely independently of Niagara by a native I/O driver. This native driver has a network interface for integration with Niagara. This also makes it possible to control the I/O of one ILC 2050 BI from the station of another ILC 2050 BI. The relevant **Address IP** settings of the Niagara driver can be accessed via the **AX Property Sheet** of the **SysmikScaloNetwork**.

The native I/O driver can be configured via **SysmikScaloNetwork / LocalPlatform**. In addition to the **Io Port**, you can also specify an **Io Remote Addr** IP address to permit remote control. The value **255.255.255.255** has a special function here: Remote control is permitted for all IP addresses.

## Station engineering

Double-click the new SysmikScaoNetwork to open the **N Device Manager**, which helps you create and manage I/O terminals. You can use the **Discover** button to initiate a search. All terminals connected to the controller are detected and displayed in the upper area of the manager. You can add I/O terminals to the station by selecting them and choosing **Add** under **Config / Drivers / SysmikScaoNetwork**.

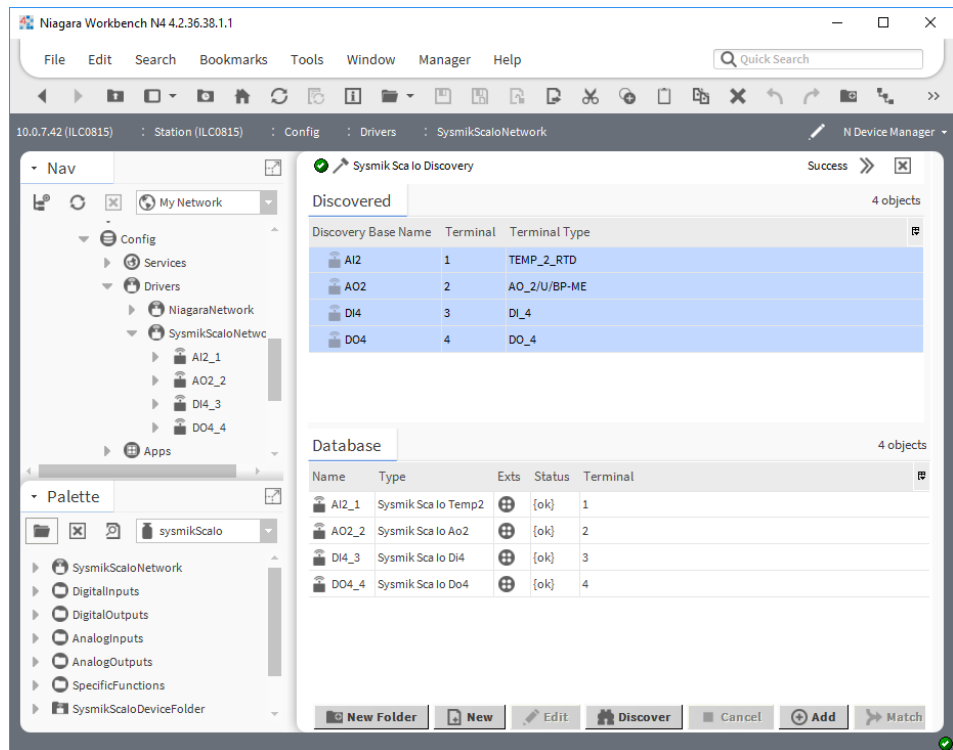



Figure 5-33 Discover search in Device Manager



You can create device folders via **New Folder** to add device components below the network component in the tree.

Double-click the Points icon  of an I/O terminal to open the **N Points Manager** of the terminal, where you can easily create and manage points. Choose the **Discover** button to display all I/O points provided by the terminal at the top of the manager. Points can be added to the station by selecting them and choosing **Add** under **Config / Drivers / SysmikScaloNetwork / <IO Terminal> / Points**.

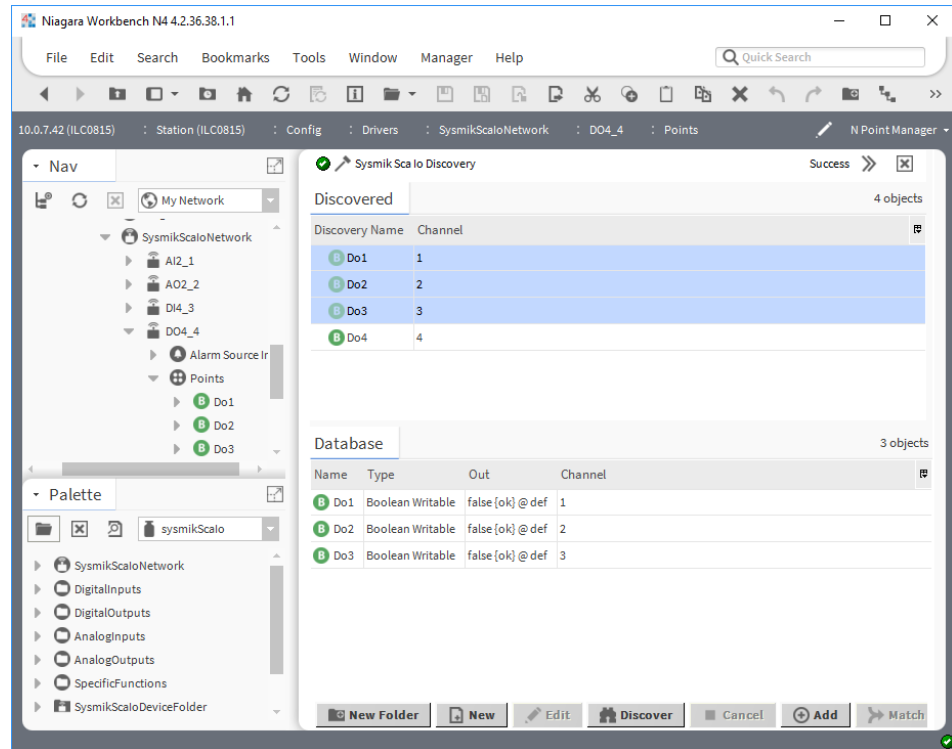


Figure 5-34 Point configuration in the Device Manager



You can create point folders via **New Folder** to add points within the points folder.

Communication in the SysmikScalo driver is event-driven. The driver cyclically communicates with the native I/O driver in the defined poll interval and automatically receives corresponding change messages when the value changes within the poll interval. For each point, one of three poll speeds can be selected via **ProxyExt / PollFrequency**. The poll cycle times for all three poll speeds can be set under **Poll Scheduler** in the **AX Property Sheet** of the **SysmikScaloNetwork**. Longer poll cycle times have no negative effect on the I/O speed due to event-driven communication.

Analog sensors can increase the processor load due to constant, negligible value changes. In this case, the ProxyExtensions of the analog inputs have an UpdateOnDelta property. A relevance threshold can be configured here. Changes during a poll cycle below this threshold are ignored.

Many analog input terminals support multiple measuring ranges. These can be configured for each affected point via the **N Point Manager** or directly via the **Ai Type** property of the **Proxy Ext** proxy extension. Frequently, electrical variables (voltage, current, resistance) are recorded by sensors that represent completely different physical parameters (pressure, volume flow, temperature). Conversion rules (linear equation,

value table, etc.) for the point can be defined via **Proxy Ext / Conversion**. You can assign suitable units to the point via **Facets**.

### 5.10.2 Serial drivers

Only Niagara controllers can communicate serially. The ILC 2050 BI has two serial RS-485 interfaces: COM1 and COM2. Additional serial interfaces can be added as needed using clip-on inline terminals. Consecutive interface identifiers COM3, COM4 ... are assigned according to the position of the terminals.



#### NOTE

Only COM1 and COM2 support BACnet MS/TP protocol.

**IB IL MBUS** terminals support the M-Bus standard and can only be used with the M-Bus communication driver.

**IB IL RS UNI** terminals support the RS-232, RS-422 and RS-485 interface standards. The interface standard can be selected via the **Rs Type** property of the corresponding **SysmikScaloRsUni** terminal component in the **SysmikScaloNetwork**.



#### NOTE

Two DIP switches may be accessible on the side of the terminal housing of the IB IL RS UNI. These must be positioned as follows before installing the terminal.

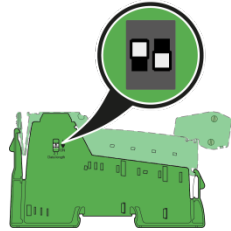


Figure 5-35 IB IL RS UNI - checking the DIP switch position

#### Example of Modbus RTU

Install the **modbusAsync-rt** module using the **Software Manager** platform tool. Double-click **Station / Config / Drivers** to open the **Driver Manager**. Choose **New** to create a new **Modbus Async Network**. The ModbusAsyncNetwork is marked as faulty in the Driver Manager. The specified reason for the error is that no serial port has been assigned: "No port selected for Modbus communication."

Open the **AX Property Sheet** view and enter interface identifier **COM1** under **Port Name** in **Property Serial Port Config**. Other parameters such as **Baud Rate** and **Stop Bits** can also be set here. The status of the network changes to {ok}.

All available serial ports with their assigned networks are listed under **Station / Config / Services / PlatformServices / SerialPortPlatformServiceNpsdk**.

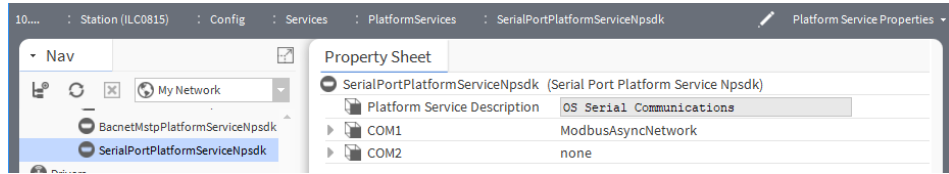


Figure 5-36 Property sheet with overview of available serial ports

## 5.11 Saving a station

### 5.11.1 Saving a station

A station is run completely from RAM. When you choose **Station / Save Station** or the **Save** action in **Station / Config**, the station, including trend and alarm data, is stored in the nonvolatile memory of the Niagara host. The station is automatically saved when the station is shut down, for example when a module is updated.

In the **Platform Service Container Plugin** default view in the **Station / config / Services / PlatformServices** platform service, you can use **Station Auto-Save Frequency** to define the interval at which the station will be automatically saved. The default is one hour. Niagara controllers such as the ILC 2050 BI use flash memory as nonvolatile memory. Flash memory can be damaged by excessive writing activity. Therefore, shortening the interval is not recommended.

The ILC 2050 BI also has an infinitely writable SRAM memory, where all changes that occur during operation are cached. In this way, no data is lost even in the event of a power failure.

If necessary, uninterruptible power supplies can be used to protect supervisor stations against loss of data.

### 5.11.2 Copying a station

Stations can be copied between Niagara hosts and the Niagara Workbench using the **Station Copier** platform tool. Trend data and alarms are also transferred in addition to the application. A station copied in this way can run on any Niagara host. There is no synchronization of version information of module versions or system software. This allows stations to be developed, tested and commissioned online on a Niagara controller or supervisor, or offline in the Niagara Workbench file system.

### 5.11.3 Backup

You can use **Station / Backup Station** to back up the station in the file system of the Niagara Workbench. This is done using a distribution file with the extension `.dist`. In addition to the application, trend data and alarms, references to installed modules and system software are also backed up. A backup can be installed in suitable Niagara hosts with the Distribution File Installer platform tool. The appropriate modules and system software are automatically installed at the same time. We strongly recommend installing a clean distribution first. A typical application is the transfer of software when devices are replaced.

## 5.12 Application structure

For the most part, the structure of the station under **Station / Config** is open. There are some conventions that simplify the maintenance of a station:

- All services are subordinate to the **Services** container service
- All drivers or network components of the drivers are subordinate to the **Drivers** driver container

Furthermore, many components have restrictions on where they can be placed in the station tree. Typically, the devices of a driver can only be placed below the corresponding network, and the points of a drive can only be placed under the corresponding devices in the **Points** folder. Alarm extensions can only be placed below points, and so on.

In addition, the user is free to decide how to structure an application. The best approach in terms of engineering effort over the product life cycle is extremely dependent on the requirements of the application. For large projects, it is usually necessary to form repetitive structures.

An I/O-based approach is recommended if the control tasks or the structure of the visualization pages follow the structure of the integrated devices. Control algorithms are created directly in the driver structure below the network or devices. One example would be monitoring / visualization of the room automation systems of an office building, with one BACnet room controller per room integrated into Niagara via a BACnet network.

Often, the application structure cannot be easily mapped to the I/O structure. For example, you may need to control the lighting control system via DALI, push buttons via Modbus and room controllers via LON and link them together. In this case, it is best to program the application functions in an I/O-independent structure and use links to connect the application functions and the I/O points.

## 6 Niagara Network

The Niagara Network is used for communication between Niagara stations. It allows you to exchange points, trend data, alarms, schedules and files as well as synchronize login credentials. In addition, you can use Provisioning to automate most system maintenance, including updates, upgrades and backups, from a central location.



### NOTE

The station name is used to identify a station in the Niagara Network. Therefore, you must ensure that station names are unique when planning a station.

### 6.1 Niagara Supervisor station

A Niagara Station can run on a Windows machine. The Niagara Supervisor software must be installed on the computer. This is referred to as a Niagara Supervisor station. Niagara Supervisor stations only support IP protocols. Since the performance of the host is potentially much higher, supervisor stations often assume a central, higher-level position in the Niagara system structure and implement the following functions, among others: web visualization, trend data archiving (also including export to external databases, if necessary) and alarm console. Communication with the lower-level stations on Niagara controllers takes place via the Niagara Network.

The Niagara platform of a Niagara Supervisor is provided by a Windows service. If this Niagara service is not already started, install and start the Niagara service from the Windows Start menu by choosing **Niagara N4 Supervisor 4. <x.x.x> Platform Daemon Installation**.

Create a new station in the Niagara Workbench by choosing **Tools - New Station** from the menu. Enter the name **Sup** in the **Station Name** field and select **NewSupervisorStationWindows.ntpl** as the station template. Assign the password **Admin12345** for the **admin** user. Select the final action, **copy it to platform for "localhost" with Station Copier**, to send the station to the Niagara platform daemon. You will be prompted to enter your user name and password in order to connect the platform. Enter your Windows login credentials here.

The Niagara Workbench automatically connects to the platform daemon. The station is transferred to the file system of the platform daemon via the **Station Copier** platform tool and started there. You can monitor the startup process via the **Application Director** platform tool, which starts automatically.

Create a station connection from the Niagara Workbench to the Supervisor Station by choosing **File - Open Station** from the menu. Select **Station TLS Connection** for the type and enter **localhost** for the host IP. You may be prompted to confirm the public key of the Supervisor. Log in with user name **admin** and password **Admin123345**.

## 6.2 Communication

The Niagara Network communicates using **Fox** protocol or the secure **Foxx** protocol based on TLS. These are the same protocols that the Niagara Workbench uses to communicate with the station. Communication can be configured via the Fox Service under **Station / Config / Services / FoxService**. By default, only Foxx is enabled on TCP port 4911. The **Foxx Cert** property can be used to select a certificate for communication.



### NOTE

The supplied **tridium** certificate can be used for the time being, but it should be replaced with a trusted certificate if secure communication is required.

If secure communication is not required, Fox protocol can be enabled via **Fox Enabled = true** and **FoxxOnly = false**. The communication parameters of the communication partners must match.

Communication is event-driven and is always established from a client to a server. Stations often assume both client and server roles for communication between two stations. This means that each communication partner must be created in the Niagara network of each station.

## 6.3 Integrating a station

Perform these steps in both stations.

Double-click **Station / Config / Drivers / NiagaraNetwork** to open the **Station Manager** of the Niagara network. Detection of stations in the network via the **Discover** button is based on UDP protocol using multicast. Under certain circumstances, settings must be made to allow for this:

- For multicast communication, a gateway address that matches the IP address must be defined in the IP settings, even if there is no gateway in the network.
- Firewalls must be configured to allow UDP communication on the Fox ports (Standard 1911 for Fox or 4911 for Foxx).

Create a device manually in the Niagara network using the **New** button.

### 6.3.1 Fox

Enter the name of the Niagara station to be integrated, **ILC0815** or **sup**, as well as the IP address of the controller or supervisor PC. Enter user name **admin** and password **Admin 12345**.



### NOTE

Administrator access should not be used in productive systems. Instead, define a network user with appropriately restricted access privileges.

Use the **Ping** action of the newly created device on the Niagara Network to verify that communication via Fox is successful. The device status must be {ok}.

### 6.3.2 Foxx

Enter the name of the Niagara station to be integrated, **ILC0815** or **sup**, as well as the IP address of the controller or supervisor PC. Set the **Fox Port** to **4911** and **Use Foxx** to

**true**. Enter user name **admin** and password **Admin 12345**.

**NOTE**



Administrator access should not be used in productive systems. Instead, define a network user with appropriately restricted access privileges.

A special feature of secure communication via TLS is the use of certificates, which a server uses to authenticate itself. These certificates must be recognized by the communication partner as trusted. Trusted sources of certificates can be maintained in the **System Trust Store** tab in the **Certificate Management** platform tool. Certificates issued by these sources are automatically trusted. The supplied Tridium certificate does not come from one of these sources and is not trusted for the time being.

**NOTE**

The supplied **tridium** certificate can be used for the time being, but it should be replaced with a trusted certificate if secure communication is required.

Use the **Ping** action of the newly created device to check whether communication is possible. The device should assume the status **{down,alarm...}**. In the **Health / Last Fail Cause** property of the device, you should see an error message about the failed certificate validation: **javax.net.ssl.SSLHandshakeException: java.security.cert.CertificateException: failed certificate validation, failed hostname validation**.

Open the **Allowed Hosts** tab in the **Certificate Management** platform tool of the client. The tab already contains an entry with the IP address of the server. The red  icon indicates that this host is untrusted. Select the line and make the host trusted using the **Approve** button, .

## Niagara Network

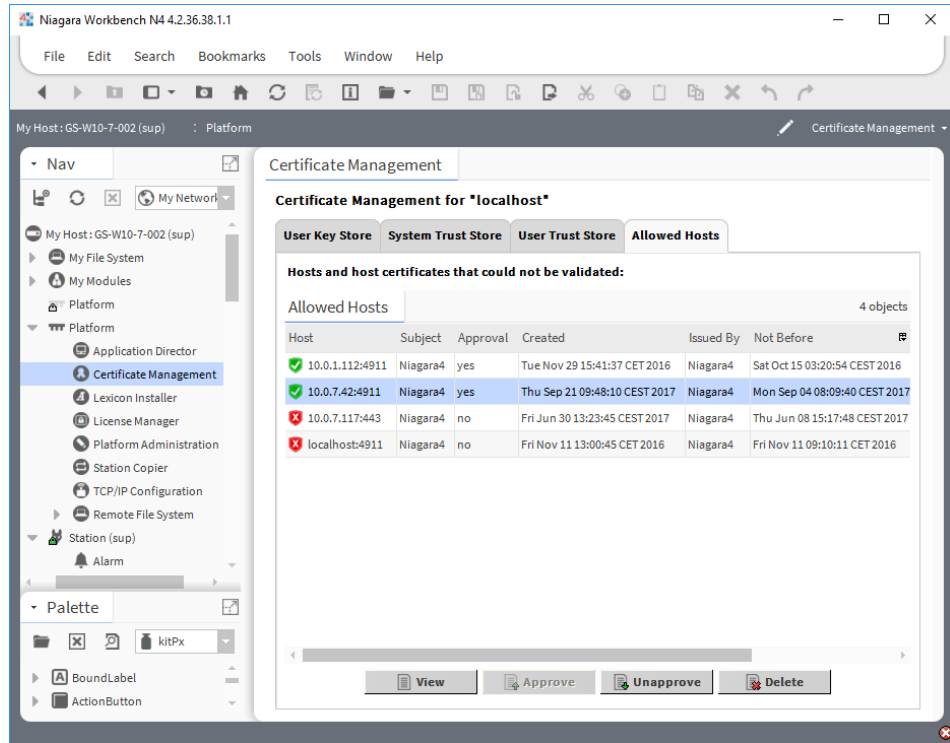


Figure 6-1 Certificate Management

Use the **Ping** action of the newly created device on the Niagara Network to verify that communication via Foxs is successful. The device status must be {ok}, or until the alarm triggered by the untrusted certificate is acknowledged, it must be {unackdAlarm}.

## 6.4 Integrating points into the Supervisor Station

Every device in the Niagara Network has device extensions. Points are integrated via the **Points** device extension. Double-click **Station (sup) / Config / Drivers / NiagaraNetwork / ILC0815 / Points** to open the **Niagara Point Manager** default view. Choose the **Discover** button to search for all **Control Point** points in the station of the ILC 2050 BI.



The Niagara Framework contains a **BQL** query language that can be used to search for information in a station using various search criteria. This query language is used to narrow the search results, but can also be used when creating a report, for example.

Double-click the **/MyAdd/Add** point to integrate it into the Supervisor Station. Once integrated, all standard functions of the point, such as alarm management, trend data collection, etc., can now be used in the Supervisor Station.



Points are read-only in the Niagara network. Due to the reciprocal integration of readable points on both communication partners, both directions of communication are still possible.

## 6.5 Integrating trend data into the Supervisor Station

Trend data is typically recorded in Niagara controllers. The Supervisor Station is better for storing data for extended periods because it has the necessary storage space.

The **Histories** device extension is used for integrating the histories of an external station. Double-click **Station (sup) / Config / Drivers / NiagaraNetwork / ILC0815 / Histories** to open the **Niagara History Import Manager**. Choose the Discover button to display all the histories of the ILC 2050 BI. Double-click the **Add** history to add it to the Supervisor Station. This creates a history in the Niagara Supervisor that is cyclically supplied with the data recorded by the ILC 2050 BI. You can define the record capacity as well as the frequency of data imports. You can also choose **Execute** to initiate the synchronization directly. The history can be reached by choosing **Station (sup) / History / ILC0815 / Add**.



Trend data can also be exported via the Niagara network. Communication takes place from the source of the data.

## 6.6 Integrating schedules into the Supervisor Station

In actual projects, it is common to integrate central schedules of a Supervisor station into lower-level Niagara stations. These schedules can be configured via a web visualization.

It also works the other way around: The **Schedules** device extension is used for integrating the schedules of an external station. Double-click **Station (sup) / Config / Drivers / NiagaraNetwork / ILC0815 / Schedules** to open the **Niagara Schedule Import Manager**. Choose the **Discover** button to display all the schedules of the ILC 2050 BI. Double-click the **NumericSchedule** schedule to add it to the Supervisor Station. Imported schedules are read-only.

## 6.7 Central alarms

In the **AlarmService** of the station triggering the alarm, a **station recipient** from the **alarm** palette is inserted and connected to the alarm class in order to route alarms of the ILC 2050 BI to the Niagara Supervisor. The **Station Recipient** has a **Remote Station** property in which the **sup** recipient station can be selected.

In the supervisor station, you can use the **Alarms** device extension to specify how incoming external alarms will be routed to internal alarm management. Double-click **Station (sup) / Config / Drivers / NiagaraNetwork / ILC0815 / Alarms** to open the **AX Property Sheet** view, where you can select the alarm class to which incoming alarms are routed.

## 7 Security

Niagara Systems use IP communication. As a result, it is very important to harden the system against potential attacks. Tridium maintains a **Niagara 4 Hardening Guide** for this purpose.

### 7.1 Passwords

Passwords are used to authenticate users. Never use default passwords. Always use secret, secure passwords that you change on a regular basis. Do not select the **Remember These Credentials** checkbox when you log in to the Niagara Workbench.

### 7.2 Login credentials

Use different login credentials for every user. Delete default login credentials (admin). Only assign permissions that are actually needed.

### 7.3 Protocols

Use secure protocols (Foxy or https). Create and maintain your own trusted certificates. Disable services or protocols that you do not need.

### 7.4 Passphrase

A system passphrase is used to encrypt sensitive data of stations such as passwords. Use a secret, secure passphrase.

### 7.5 Environment

Install Niagara systems in secure environments. This means installing them in access-controlled rooms or cabinets. This obviously applies to the network infrastructure as well. Protect the Niagara against attacks from the Internet by using a VPN.

### 7.6 ILC 2050 BI

The ILC 2050 BI contains functions that you should disable via the commissioning website if you are not using them. This applies to the Sedona Virtual Machine as well as the I/O remote control.

## 8 Niagara diagnostics

Whenever a technical system is not behaving as expected, the causes must be determined and eliminated.

### 8.1 Version information

Prior to contacting Support, version information should be noted and evaluated.

In the **Platform Administration** platform tool, you can view all version information by choosing the **View Details** button. You can then choose the **Copy to Clipboard** button to save the information to the clipboard for e-mailing.

If any software updates or upgrades have been released, they should be installed. Maintenance and support for older versions expires when a new version of Niagara software is released.

### 8.2 Event logging

All Niagara modules can display diagnostic messages. For this purpose, modules have loggers that typically use the same name as the module or a name based on it. Every diagnostic message has a log level to classify the severity of the message from **FINEST** (detailed information not required for operation) to **SEVERE** (serious errors). For each logger, a log level can be specified to indicate the minimum severity level at which diagnostic messages will be output via the station output. For most loggers, this level is **INFO**. This means that all diagnostic messages with the log levels **INFO**, **WARNING** and **SEVERE** are displayed.

Many modules display highly detailed diagnostic messages with a very low log level. For drivers, these messages may even include protocol records.

#### 8.2.1 Niagara station

You can double-click the default view of the **DebugService** under **Station / Config / Services** to open the **Logger Configuration** view. Here you can define a default level (**ROOT**) for all loggers and deviations from this default level for individual loggers.

## Niagara diagnostics

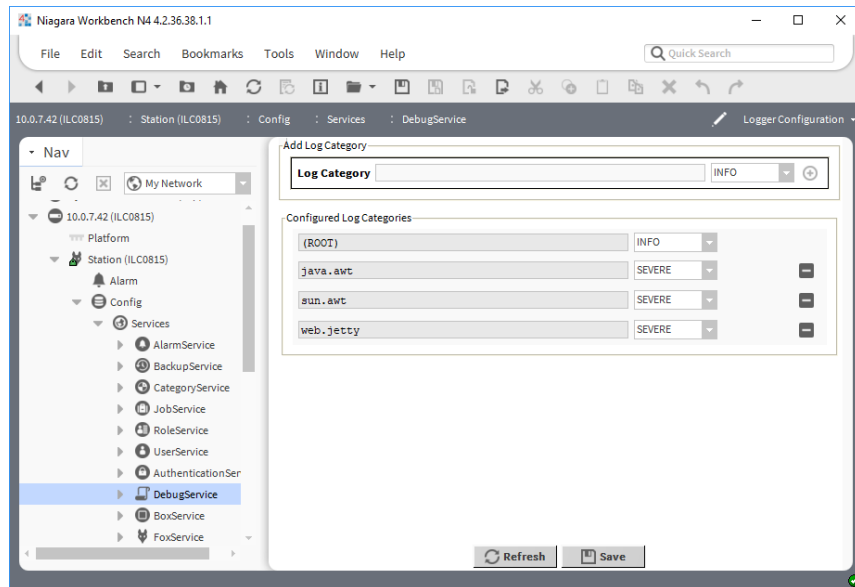


Figure 8-1 Station logger configuration

Messages are output via the station output in the **Application Director** platform tool.

Additionally, the **LogHistoryService** stores events in the **LogHistory** of the station, which can be viewed by choosing **Station / History / <Station Name> / LogHistory**.

The **AuditHistoryService** logs changes to the station in the **AuditHistory**, which can be viewed by choosing **Station / History / <Station Name> / AuditHistory**.

### 8.2.2 Niagara Workbench

The Niagara Workbench is used for engineering a station. Components of the user interface such as views, managers and plug-ins are not executed in the station, but directly in the Niagara Workbench. When irregularities appear, it can also be useful to display diagnostic messages for these software components. To do this, start the Niagara Workbench with the console: **Workbench N4 (Console)** start menu.

You can also choose **Tools - Logger Configuration** to display the same **Logger Configuration** view that is used for the DebugService of the Niagara Station (8.2.1). However, this view is specifically for the Niagara Workbench. In other words, you use this view to define the diagnostic messages that the Niagara Workbench will display in the external console.

### 8.3 Execution of applications

The Niagara Station consists of components whose code is executed by different mechanisms.

Many components that only have processing functions (such as arithmetic operations) calculate their output values only when changes are made to their own inputs.

Components with time functions (for example signal generators or PID controllers) are also executed cyclically at intervals that can usually be configured.

Depending on the protocol, outputs of proxy points of the communication drivers are

recalculated when a corresponding protocol frame is received or read by the **Poll Scheduler**. Only points that are actually needed at the moment are read, meaning their values are used for other tasks such as displaying an active view, issuing alarms, recording trend data or performing calculations. Writing, on the other hand, is typically event-driven and occurs when a value changes.

All events are decoupled by queues. Event-driven processing is highly effective. However, there is no fixed cycle time and no reliable information on system responsiveness.

As a guideline, the processor load (**Resource Manager** view of the **station**) should not exceed 80%. The processor load can be reduced by extending execution cycles / poll cycles and reducing the number of value changes.

## 8.4 System diagnostics - Spy

The Niagara Framework includes system diagnostics functions that are consolidated in the **Spy** local menu of the connected station. Only a few of these functions will be mentioned here.

### 8.4.1 EngineManager

You can obtain information about the processing of the application by choosing **sysManagers - engineManager**. When processor load is high, you can choose **Engine Hogs** to identify components that are taking up a disproportionately large amount of computing time.

### 8.4.2 ClassLoaders

You can choose **classLoaders** to display a list of all installed modules. This list also indicates the number of installed elements of these modules as well as the amount of memory they are using. Modules consuming a disproportionately large amount of memory can be easily identified here.

### 8.4.3 Log setup

You can choose **logSetup** to display an alternative view for setting up the logger configuration (8.2.1).

## 9 Software maintenance

All components of the Niagara Framework are versioned. This includes the system software as well as all Niagara modules. Updates within a Niagara minor version as well as upgrades to a higher Niagara minor version are delivered via new installations of the Niagara Supervisor and the SysMik Niagara AddOns. Note the following with regard to licensing for upgrades: When a new Niagara minor version is released, licenses are only automatically applied to the new version for the duration of software maintenance period.

### 9.1 Niagara Workbench

When Niagara Supervisor is installed, the Niagara version is part of the proposed installation path. In this way, multiple versions of the Niagara Workbench can be operated independently on a single PC. SysMik Niagara AddOns are always related to a Niagara version and can only be installed in the installation directories of corresponding supervisor versions.

### 9.2 Supervisor

Software for the Niagara Supervisor is not installed via a commissioning process in the Niagara Workbench, but directly via the Niagara Supervisor installation. A Niagara Supervisor and Niagara Workbench with the same version use the same Niagara modules.

Only one Niagara platform daemon can be active on a computer. The installation of the Windows service can be activated during the Supervisor installation or later via the **Platform Daemon Installation** start menu. The installation will automatically stop and uninstall an already active platform daemon from another Niagara version.

When the Niagara software is updated or upgraded, the Niagara Supervisor Station can be copied back from the Niagara Supervisor to the Niagara Workbench using the **Station Copier** platform tool, and after the new Platform Daemon is installed, it can be copied back from the Niagara Workbench to the new Niagara Supervisor.

### 9.3 IB IL 2050 BI

Updates and upgrades should always be installed through a commissioning process. Individual updates to modules can also be installed with the **Software Manager** platform tool.

If you want to downgrade the system software, install the appropriate clean distribution for the Niagara host by choosing the **Cleaning** button in the **Distribution File Installer** platform tool. This is a minimal installation from which a Niagara version can be installed using the Commissioning Wizard.

---

## Please observe the following notes

### **General terms and conditions of use for technical documentation**

Phoenix Contact reserves the right to alter, correct, and/or improve the technical documentation and the products described in the technical documentation at its own discretion and without giving prior notice, insofar as this is reasonable for the user. The same applies to any technical changes that serve the purpose of technical progress.

The receipt of technical documentation (in particular user documentation) does not constitute any further duty on the part of Phoenix Contact to furnish information on modifications to products and/or technical documentation. You are responsible to verify the suitability and intended use of the products in your specific application, in particular with regard to observing the applicable standards and regulations. All information made available in the technical data is supplied without any accompanying guarantee, whether expressly mentioned, implied or tacitly assumed.

In general, the provisions of the current standard Terms and Conditions of Phoenix Contact apply exclusively, in particular as concerns any warranty liability.

This manual, including all illustrations contained herein, is copyright protected. Any changes to the contents or the publication of extracts of this document is prohibited.

Phoenix Contact reserves the right to register its own intellectual property rights for the product identifications of Phoenix Contact products that are used here. Registration of such intellectual property rights by third parties is prohibited.

Other product identifications may be afforded legal protection, even where they may not be indicated as such.

---

## How to contact us

**Internet**

Up-to-date information on Phoenix Contact products and our Terms and Conditions can be found on the Internet at:

[phoenixcontact.com](http://phoenixcontact.com)

Make sure you always use the latest documentation.

It can be downloaded at:

[phoenixcontact.net/products](http://phoenixcontact.net/products)

**Subsidiaries**

If there are any problems that cannot be solved using the documentation, please contact your Phoenix Contact subsidiary.

Subsidiary contact information is available at [phoenixcontact.com](http://phoenixcontact.com).

**Published by**

PHOENIX CONTACT GmbH & Co. KG

Flachmarktstraße 8

32825 Blomberg

GERMANY

Should you have any suggestions or recommendations for improvement of the contents and layout of our manuals, please send your comments to:

[tecdoc@phoenixcontact.com](mailto:tecdoc@phoenixcontact.com)



# SCATTERGOOD & JOHNSON LTD

ELECTRICAL ENGINEERING & FLUID CONTROL DISTRIBUTORS

Est.1899

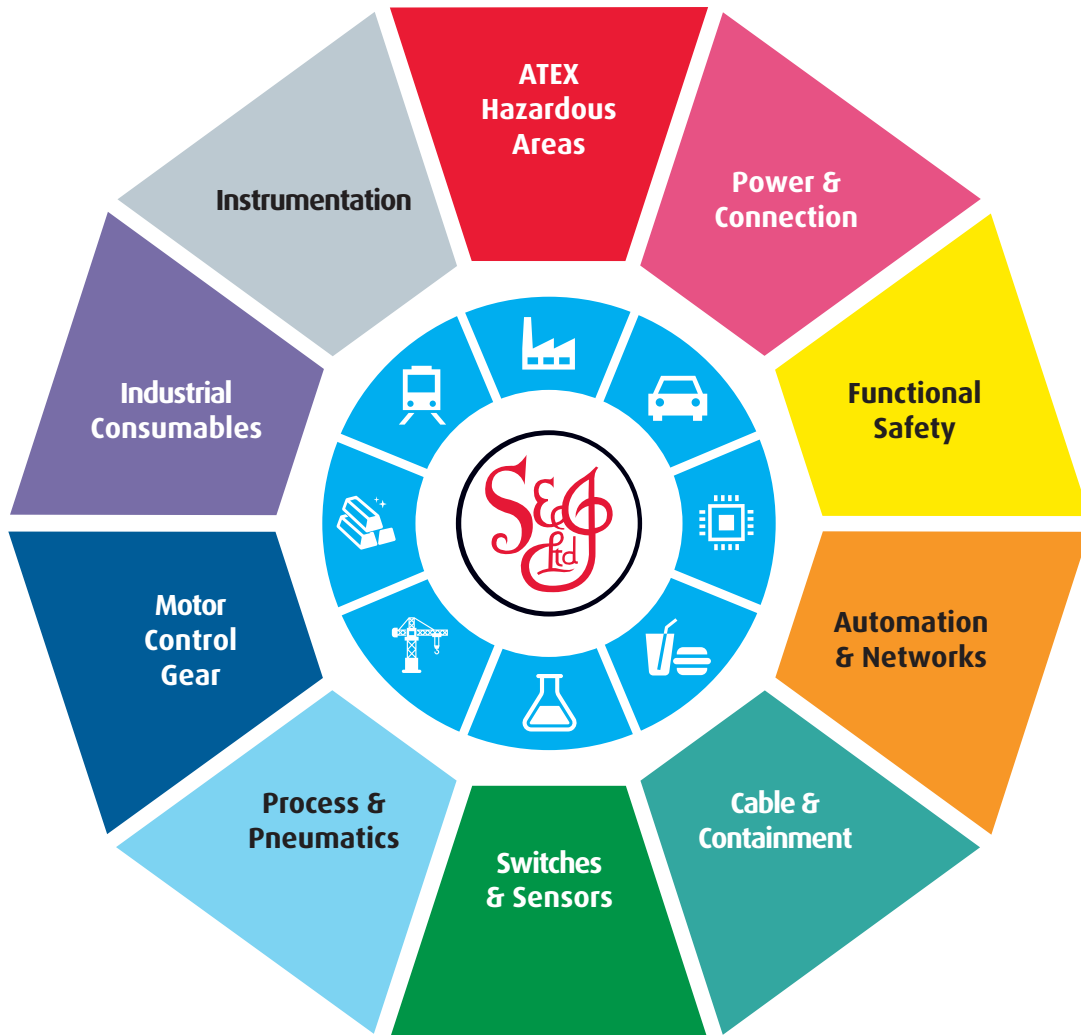
At Scattergood & Johnson Ltd, we pride ourselves on being a technical distributor to specialist industries.

Working with a range of quality product suppliers across a number of specialist markets, we are not your average 'box shifter' - we are your technical and supply chain partner.

We fully support every product we sell - for free! Our internal team and external sales engineers can answer any product or application question, no matter the complexity.

Backing up this technical ability is a range of 50,000+ products available from stock for nationwide next day delivery (same day if required!), or you can collect what you need from any of our trade counters around the UK.

Select your specialist interest below to learn more about how we can help.



Online, In Branch and On the Road - Scattergood & Johnson Ltd, there when you need us.

# [www.scatts.co.uk](http://www.scatts.co.uk)