# Schneider Electric ION Meter Driver for Tridium Niagara

User Guide

## Contents

# 1 Introduction to ION

ION electrical meters were originally developed by Power Measurement Inc[1], which later became part of Schneider Electric family. All ION meters are built on so-called ION architecture – *Integrated Object Network* – a very flexible and powerful software framework. Because of this framework, in many aspects ION devices are closer to programmable logical controllers than to meters.

ION family consists of many models (in order of complexity):

| | | | | | |
|---|---|---|---|---|---|
| ION6200 | ION7300 | ION7330 | ION7350 | ION7400 | ION7500 |
| ION7550 | ION7600 | ION7650 | ION7700 | ION8300 | ION8400 |
| ION8500 | ION8600 | ION8650 | ION8800 | PM8000 | ION9000 |

Thanks to ION architecture one ION device simultaneously can operate as:

- electrical meter
- power analyzer
- programmable logical controller
- serial-IP interface
- protocol gateway

As one might expect, ION meters do measure electrical network parameters with high precision (up to Class 0.1).

ION meters constantly monitor the quality of electrical voltage and current to detect possible power disturbances: undervoltage, overvoltage, sag/swell, harmonic distortion, transients. Although many meters provide some power quality metrics, e.g. total harmonic distortion or THD, ION meters do it in a very detailed way. For example, they calculate voltage and current harmonics up to 127th order and for sag/swell and transient analysis they capture the measured signal waveform as often as 30,000 times per second. Power quality analysis can be performed according to EN50160 and IEC 61000-4-30 Class A standards.

ION meters can be equipped with digital and analog inputs and outputs, which can be monitored and controlled in PLC-like fashion. It is possible to build complex relay switching logic or even implement a PID loop in ION.

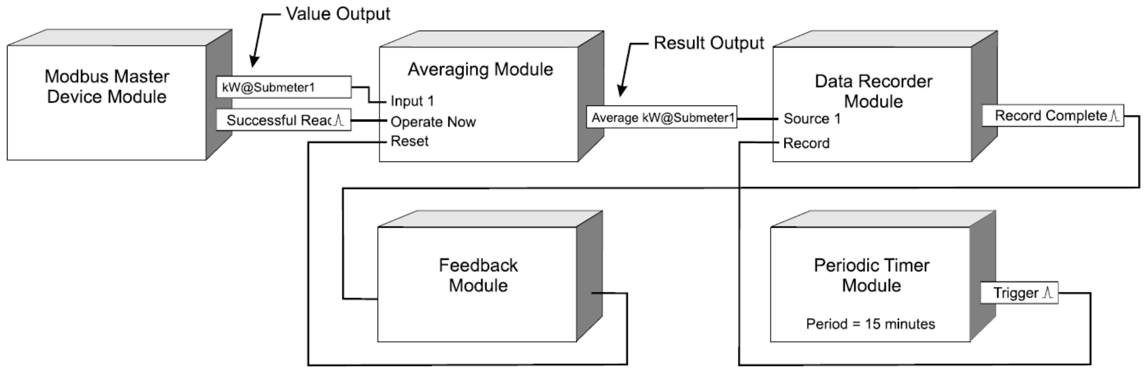
ION meters can integrate multiple protocols to send and receive information. Their main protocol is always the native ION available in IP and RS-485 versions. Various ION models support serial protocols (Modbus RTU, DNP3, LonWorks, Profibus DP) and Ethernet protocols (Modbus TCP, DNP3 TCP, IEC 61850, DLMS/COSEM). They can store captured waveforms in COMTRADE format on built-in SFTP server, send alarms to SNMP traps or to e-mails, synchronize time via NTP, PTP or GPS, create redundant network loop using RSPT (Rapid Spanning Tree Protocol). Serial devices with ION or Modbus protocol working via RS-485 can be connected to IP network via ION EtherGate feature.

ION meters are the key element of EcoStruxure Power, Schneider Electric integral system for electrical installations. They are often accompanied with Power Monitoring Expert (also known as PME), a specialized SCADA system. Although ION meters can communicate with multiple protocols, their full functionality can only be accessed with proprietary ION protocol. Until now PME was the only software which can ‘speak’ ION and utilize the full power of ION devices. Now their power is open to Tridium Niagara.

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

## 2 ION Architecture

ION framework is a combination of various functional blocks, which are linked to each other, similar to Niagara wiresheets.



**ION terminology**

- *Node* – a device, either physical (ION meter) or software one (Virtual Processor module in PME).
- *Module* – a building block of ION architecture, similar to Niagara component. Each module implements some function, from simple (Maximum, AND/OR) to complex ones (fast Fourier transform). Modules might have inputs, outputs and setup registers. If inputs are linked to other module outputs, then their values are processed and the results are sent to module outputs. These outputs can be linked to other module inputs. Module logic can be configured with setup registers.
- *Framework* – a group of ION modules that are linked together and configured to perform a specific function. For example, the *Power Quality* framework can monitor disturbances such as voltage sags and transients, analyze surges and monitor real-time harmonics.
- *Template* – the whole device program, composed of multiple frameworks. Every ION device is shipped with a factory-configured template.
- *Manager* – a directory of modules of a certain type available in the node. For example, *Maximum Modules* manager stores all Maximum modules available in this device, which can be used in various frameworks.

ION modules can be

- *Core* – fixed at factory and cannot be added or removed, e.g. *Power Meter* module.
- *Standard* – all other modules, which can be used for custom logic, e.g. *AND/OR* modules.

ION meters communicate with ION protocol transmitted via one of two physical interfaces: *ION TCP* (TCP/IP on Ethernet) or *ION serial* (RS-485 bus). ION TCP devices are equipped with RS-485 ports, which can be set to “EtherGate” mode allowing to connect ION serial devices to IP network.

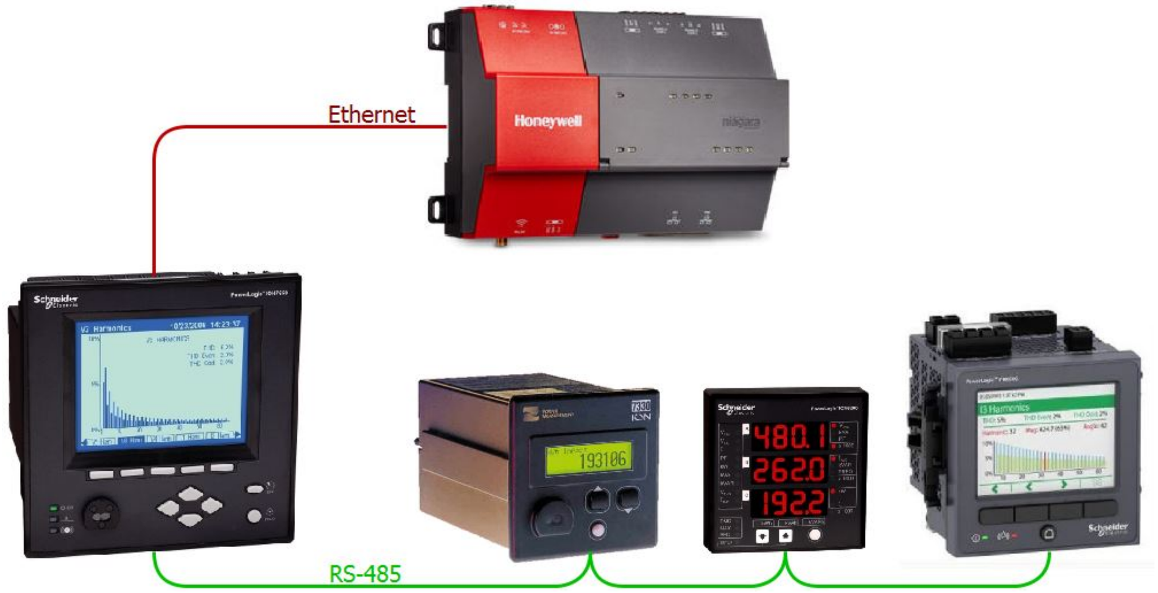
For a comprehensive guide to ION technology and all types of modules see ION Reference.

## 3 Niagara Driver

ION driver for Niagara is available on both Niagara AX and Niagara 4 platforms. The driver communicates with ION meters via IP network. It can also reach RS-485 meters[2] connected to

[2]If support for serial (RS-485) ION protocol is required, please e-mail us.

IP-enabled meter COM port set to EtherGate mode.



The driver can discover all ION managers, modules, their inputs / outputs / setups registers and map them into Niagara points of appropriate type. When possible, the driver assigns point precision and physical units. Then it automatically combines these points in groups and polls them as often as required. ION data logs (values captured with seconds-minutes-hours intervals) and waveforms (instantaneous values captured with microseconds-milliseconds intervals, used to analyze energy discrepancies) are periodically imported as Niagara histories.

All information received from ION meters can then be used to build front-end graphics, implement custom logic, export to other protocols or send to the cloud.

## 3.1 Requirements

- Niagara-powered device with software v3.8 (AX), v4.1 (N4) or later, including Jace 3, Jace 6, Jace 8000, Supervisor and their OEM and EDGE versions
- ION driver module and license

## 3.2 Quick Start

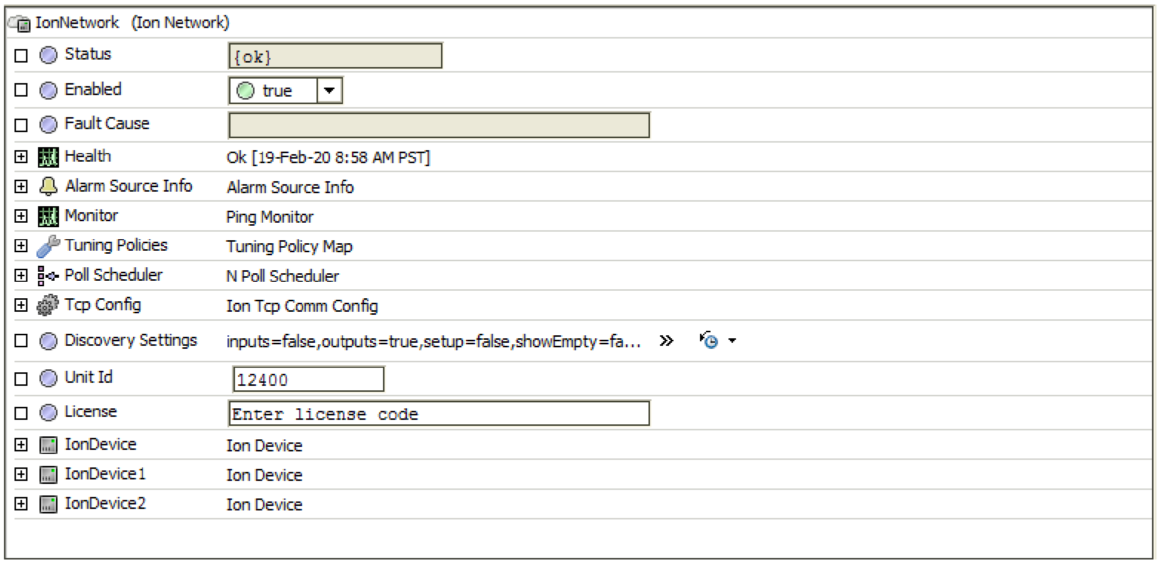
1. Import ion.jar (AX) or ion-rt.jar (N4) module with all dependencies into both Jace and WorkPlace, restart both.
2. Add **Ion Network** to the station.
3. Enter license code in **License** property under **Ion Network**.
4. Add new **Ion Device** and enter its IP address, port and unit ID. Right-click and ping to check communication – meter model should appear.
5. Open device **Points** extension and **Discover** points.
6. Add points to the station.

## 3.3 ION Network

**Ion Network** contains many standard Niagara properties, as well as few driver-specific ones:

- **Discovery Settings** – point discovery option, see **ION Points**
- **Unit ID** – ION unit address assigned to Niagara; should not be changed
- **License** – the code which allows driver to run on your host

Normally only **License** property should be changed.



IonNetwork (Ion Network)

| | |
|---|---|
| Status | {ok} |
| Enabled | true |
| Fault Cause | |
| Health | Ok [19-Feb-20 8:58 AM PST] |
| Alarm Source Info | Alarm Source Info |
| Monitor | Ping Monitor |
| Tuning Policies | Tuning Policy Map |
| Poll Scheduler | N Poll Scheduler |
| Tcp Config | Ion Tcp Comm Config |
| Discovery Settings | inputs=false,outputs=true,setup=false,showEmpty=fa... |
| Unit Id | 12400 |
| License | Enter license code |
| IonDevice | Ion Device |
| IonDevice1 | Ion Device |
| IonDevice2 | Ion Device |

## 3.4 ION Devices

ION protocol has no means to automatically discover ION devices, so addresses for each meter should be manually collected either from meter display, PME software or supporting documentation.

ION meters can be connected to:

- Ethernet LAN via RJ45 port (ION TCP)
- RS-485 bus (ION serial), which terminates at COM port of IP-enabled meter with EtherGate mode enabled
- RS-485 bus (ION serial), which terminates at COM port of serial-IP interface like Schneider EGX150

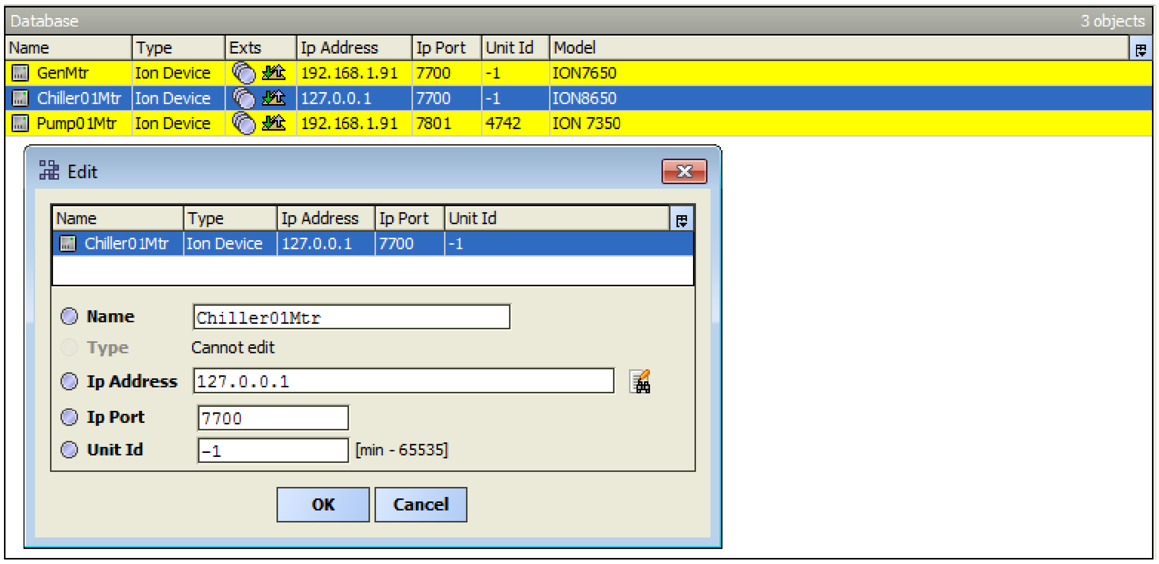
Each device is identified with three addresses: IP address, TCP port and unit ID. IP address is either IP address of ION meter itself or IP address of EtherGate gateway device. TCP port is either 7700 for ION TCP devices or 7801-7803 for RS-485 devices connected to EtherGate gateway (7801 - COM1 port, 7802 - COM2 port, 7803 - COM3). Unit ID is a number from 1 to 9999 to identify RS-485 meters. For IP meters is should be -1.

Each **Ion Device** has the following properties:

- **Poll Frequency** – used for point polling, see **ION Points**
- **IP Address** – IP address for the device itself (for ION TCP devices) or gateway IP address
- **TCP Port** – TCP port for ION TCP device (7700) or EtherGate port (7801-7803)
- **Unit Id** – serial address for RS-485 device (1-9999)
- **Model** – device model

Each **Ion Device** has **Points** and **Histories** extensions.

Device is added to the station by clicking **New** button and filling requested properties. After the device is added the driver will try to establish communication by reading device model, which will appear in **Model** column. If the model is not shown and the device color becomes yellow (offline status), then the addresses and communication should be checked, e.g. with computer **ping** command.



Database 3 objects

| Name | Type | Exts | Ip Address | Ip Port | Unit Id | Model |
|---|---|---|---|---|---|---|
| GenMtr | Ion Device | | 192.168.1.91 | 7700 | -1 | ION7650 |
| Chiller01Mtr | Ion Device | | 127.0.0.1 | 7700 | -1 | ION8650 |
| Pump01Mtr | Ion Device | | 192.168.1.91 | 7801 | 4742 | ION 7350 |

Edit

| Name | Type | Ip Address | Ip Port | Unit Id |
|---|---|---|---|---|
| Chiller01Mtr | Ion Device | 127.0.0.1 | 7700 | -1 |

| | |
|---|---|
| Name | Chiller01Mtr |
| Type | Cannot edit |
| Ip Address | 127.0.0.1 |
| Ip Port | 7700 |
| Unit Id | -1 [min - 65535] |

## 3.5 ION Points

ION data model consists of managers (collection of commands), modules (commands) and registers (points). Each of them has a unique address called **Handle ID**. They can be automatically discovered by clicking **Discover** button. Managers and modules are shown as folders and registers as points. ION devices often contain a lot of registers (in order of thousands and more), so the discovery process might take up to few minutes on RS-485 devices – its status could be monitored in a progress bar. To minimize discovery time and simplify displayed data structure, only output registers are discovered by default. Input registers do not add extra information as one module's input is another module's output. Setup registers are static and do not change with time. Still, if necessary, input and setup registers can be discovered by enabling appropriate modes in **Ion Network / Discovery Settings** facets. It is also possible to display empty managers and modules, which are hidden by default.

After all required points are discovered, they can be added to the station by selecting them and clicking on **Add** button. The points type (Numeric, Boolean, String) will be automatically selected and, where possible, numeric precision and physical units will be assigned.

ION registers do have understandable names and their actual values are displayed during discovery, so it is possible to identify points of interest. Still, the number and the variety of modules is very high, so here is an short description of some useful managers:

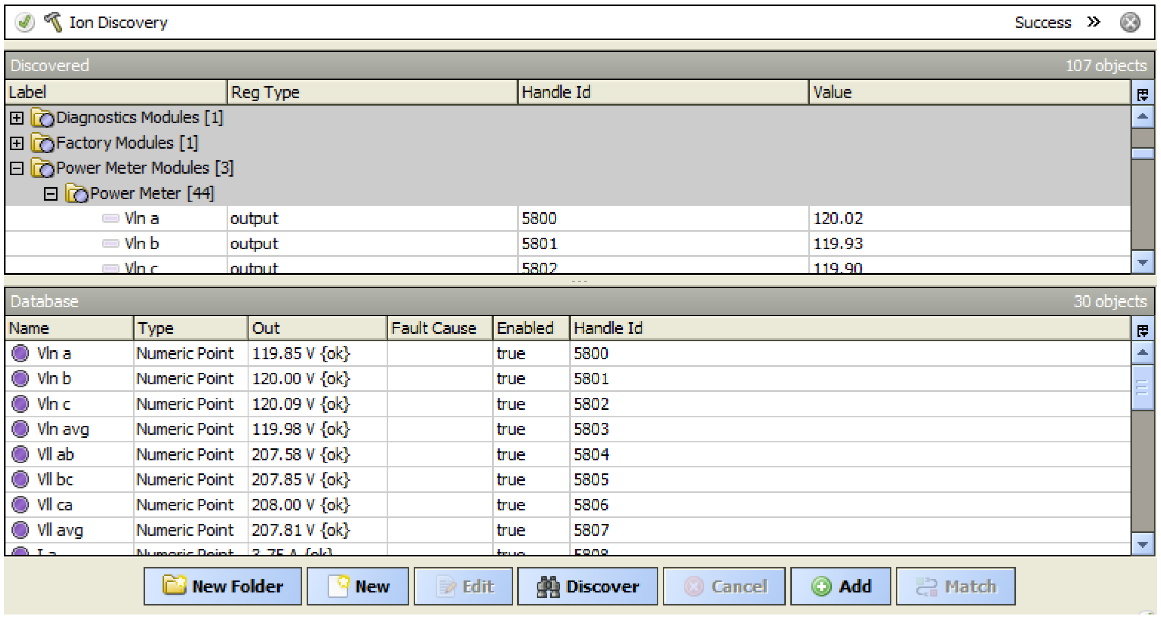
- **Power Meter Modules** – all basic electrical values: voltage, current, power
- **Integrator Module** – energies and other integrated values
- **Harmonics Modules** – amplitudes of harmonics (up to 128th for top models) as well as total harmonic distortion and phasor values
- **Analog / Digital / Counter In / Out Modules** – status of meter inputs and outputs
- **Sag/Swell Modules and Transient Modules** – summary of last sag / swell / transient distortions

- **Minimum / Maximum / Averaging Modules** – min/max/avg calculated values
- **Arithmetic Module** – various data calculated using custom formulas
- **SWinDemand Modules** – values calculated with sliding time window
- **Disturbance Analyzer Modules** – information about flicks, dips, interrupts, overvoltage
- **Diagnostics Modules** – technical info about the meter: available memory, cycle times
- **Data Rec and Wform Rec** – logs, see **ION Logs** **section
- **External Boolean / Numeric / Pulse / String Modules** – constant values

There are much more types of modules are available and their full description is given in ION Reference and all meter templates are described in ION Device Template Reference.

ION meters with identical templates have identical managers, modules and registers, which allows to fully configure only one meter of certain type (add all required points and histories, maybe alarms and relativized graphics), then duplicate it as many times as necessary and change names and addresses to match other meters. This way system engineering time will be minimized.

All added points are polled by driver driver in groups to minimize the traffic. Polling rate is specified in **Ion Device / Poll Frequency** property.



Discovered 107 objects

| Label | Reg Type | Handle Id | Value |
|---|---|---|---|
| ⊞ Diagnostics Modules [1] | | | |
| ⊞ Factory Modules [1] | | | |
| ⊟ Power Meter Modules [3] | | | |
| ⊟ Power Meter [44] | | | |
| Vln a | output | 5800 | 120.02 |
| Vln b | output | 5801 | 119.93 |
| Vln c | output | 5802 | 119.90 |

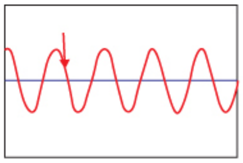
Database 30 objects

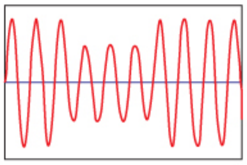
| Name | Type | Out | Fault Cause | Enabled | Handle Id |
|---|---|---|---|---|---|
| Vln a | Numeric Point | 119.85 V {ok} | | true | 5800 |
| Vln b | Numeric Point | 120.00 V {ok} | | true | 5801 |
| Vln c | Numeric Point | 120.09 V {ok} | | true | 5802 |
| Vln avg | Numeric Point | 119.98 V {ok} | | true | 5803 |
| Vll ab | Numeric Point | 207.58 V {ok} | | true | 5804 |
| Vll bc | Numeric Point | 207.85 V {ok} | | true | 5805 |
| Vll ca | Numeric Point | 208.00 V {ok} | | true | 5806 |
| Vll avg | Numeric Point | 207.81 V {ok} | | true | 5807 |
| I a | Numeric Point | 3.75 A {ok} | | true | 5808 |

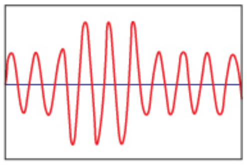
## 3.6 ION Histories

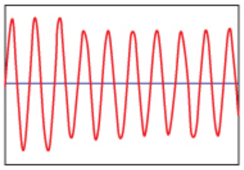
One of the most advanced features of ION meters is the ability to function as oscilloscope – capture electrical signals with very high frequency and store them in memory as logs for future analysis. There are two types of logs in ION:

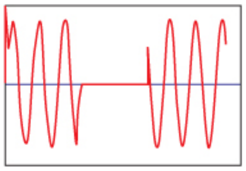
- **Data Logs** are stored continuously with preset interval in order of seconds-minutes-hours or by demand. These logs are very similar to Niagara histories.
- **Waveform Logs** are captured very fast – up to 30,000 times per second. They are commonly used to capture voltage and current signals during sag/swell or transient disturbances. The capture starts few periods before the event and continues for a defined number of periods, thus allowing to see the disruption in all details and come up with informed decision about its reasons.

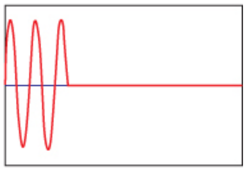

(*a*) Transient

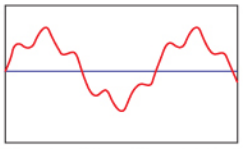

(*b*) Sag

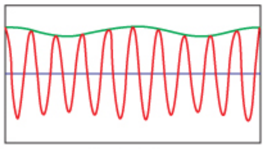

(*c*) Swell

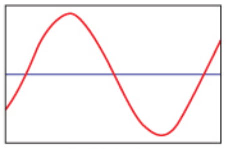

(*d*) Undervoltage


(*e*) Momentary interruption


(*f*) Long term interruption


(*g*) Steady state voltage distortion


(*h*) Flicker

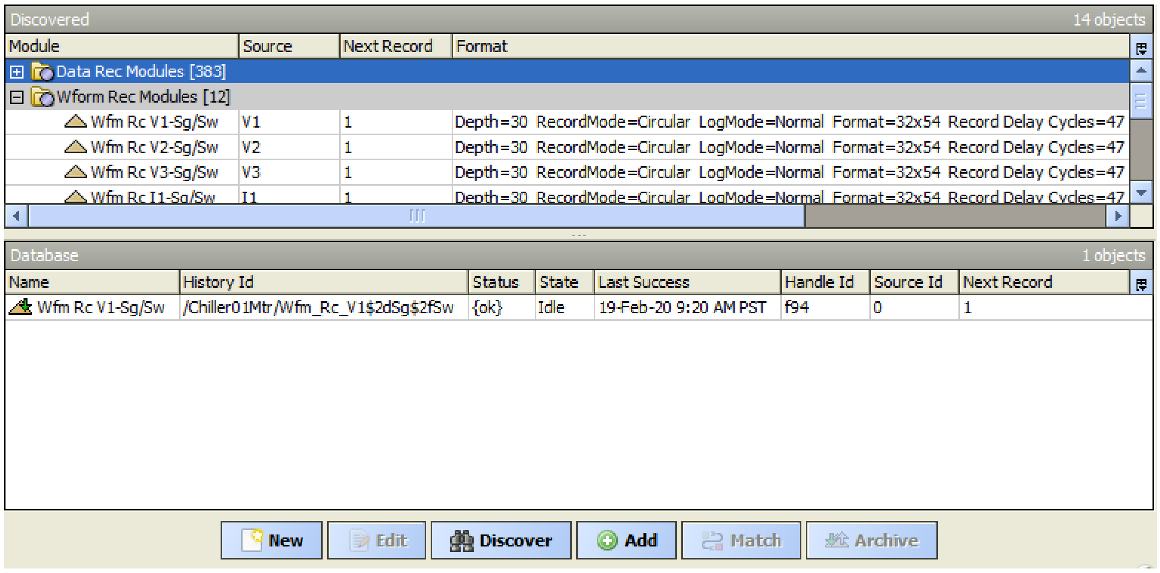

(*i*) Noise

ION logs are captured by recorder modules: **Data Rec** and **Wform Rec**. Each **Data Rec** can have up to 16 sources and collect up do 16 logs at the same time. **Wform Rec** has just one source (e.g. voltage phase 1) and it collects many waveforms.

Logs can be automatically discovered in **Histories** extension by pressing **Discover**. For all found logs the driver displays:

- **Module** – log module name
- **Source** – name of input variable
- **Next Record** – the pointer of the next record. If it is equal to 0, then no records are collected yet
- **Format** – various properties, the most important ones are **Depth** (maximum number of stored records / waveforms), waveform **Format** (number of samples per period x number of periods)
- **Handle Id** – module address
- **Source Id** – input address in the module

ION logs can be added to the station by selecting them and clicking on **Add** button. The driver will create **History Import** object, which produce Niagara histories when the **Archive** action is activated either manually (by pressing of **Archive** button or selecting **Execute** action) or automatically by **Execution Time** trigger (at preset day time or after certain interval). When archiving logs for the first time, it may take awhile, because the driver will collect all available records / waveforms. Next time the driver will only read new records, so it will be much faster.


Discovered 14 objects

| Module | Source | Next Record | Format |
|---|---|---|---|
| ⊞ Data Rec Modules [383] | | | |
| ⊟ Wform Rec Modules [12] | | | |
| Wfm Rc V1-Sg/Sw | V1 | 1 | Depth=30 RecordMode=Circular LogMode=Normal Format=32x54 Record Delay Cycles=47 |
| Wfm Rc V2-Sg/Sw | V2 | 1 | Depth=30 RecordMode=Circular LogMode=Normal Format=32x54 Record Delay Cycles=47 |
| Wfm Rc V3-Sg/Sw | V3 | 1 | Depth=30 RecordMode=Circular LogMode=Normal Format=32x54 Record Delay Cycles=47 |
| Wfm Rc I1-Sg/Sw | I1 | 1 | Depth=30 RecordMode=Circular LogMode=Normal Format=32x54 Record Delay Cycles=47 |

Database 1 objects

| Name | History Id | Status | State | Last Success | Handle Id | Source Id | Next Record |
|---|---|---|---|---|---|---|---|
| Wfm Rc V1-Sg/Sw | /Chiller01Mtr/Wfm_Rc_V1$2dSg$2fSw | {ok} | Idle | 19-Feb-20 9:20 AM PST | f94 | 0 | 1 |

**Note:** in the present version waveform histories are stored with modified timestamps: the difference between samples is always 1 second, although in reality it could be less than 1 millisecond. This is due to limitations of Niagara chart widget, which does not work correctly with milli- or microseconds. This approach allows to evaluate waveform shape in standard Niagara chart viewer. We are working on a better solution.