



**Pacific Northwest**  
NATIONAL LABORATORY

*Proudly Operated by **Battelle** Since 1965*

# VOLTRON™ Device Driver Configuration

ROBERT LUTES  
KYLE MONSON

Pacific Northwest National Laboratory

VOLTRON™ 2017



# VOLTTRON™ Driver Resources

- ▶ Current documentation for configuring and using VOLTTRON™ drivers (BACnet and Modbus) can be found at:
  - [http://volttron.readthedocs.io/en/master/core\\_services/drivers/index.html](http://volttron.readthedocs.io/en/master/core_services/drivers/index.html)

# Automatically Generating BACnet Configuration Files



Pacific Northwest  
NATIONAL LABORATORY

*Proudly Operated by* **Battelle** *Since 1965*

- ▶ This talk will walk through the current device driver configuration documentation on Read the Docs to:
  - Perform network discovery of BACnet devices
  - Scrape configuration files for these devices
  - Start trending for these devices in VOLTTRON™
  
- ▶ [http://voltron.readthedocs.io/en/master/core\\_services/drivers/BACnet-Auto-Configuration.html](http://voltron.readthedocs.io/en/master/core_services/drivers/BACnet-Auto-Configuration.html)



# Network Discovery of BACnet Devices

- ▶ What the network discovery script does:
  - Shows a list of BACnet devices that are on a network and responding to BACnet services (e.g., Whols, IAm).
  - Gives the network address (IP or remote station) and BACnet device instance. These parameters are needed to generate a BACnet registry configuration file.
  
- ▶ What the network discovery script does not do:
  - Identify what type of devices are on the network:
    - The network discovery script does not tell what the devices are or give specific device information (i.e., the script cannot discern whether the device is an AHU controller, VAV controller, or some other type of BACnet device).
    - Information related to the devices on the control system is useful and usually necessary (especially for buildings/campuses with many BACnet devices) for configuration of drivers for data trending.

# Configuration of the Network Discovery Script

- ▶ 1. Ensure the VOLTTRON™ environment is activated.
- ▶ 2. Configure the BACpypes.ini file:
  - Get your network interface information
    - In Linux this can be obtained using the command:

- ◆ ifconfig

```
(volttron)volttron@volttron:~/volttron$ ifconfig
enp0s3  Link encap:Ethernet  HWaddr 08:00:27:8d:ec:91
        inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
        inet6 addr: fe80::a00:27ff:fe8d:ec91/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:75583 errors:0 dropped:0 overruns:0 frame:0
        TX packets:30017 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:85179054 (85.1 MB)  TX bytes:2266163 (2.2 MB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:483 errors:0 dropped:0 overruns:0 frame:0
        TX packets:483 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1
        RX bytes:39283 (39.2 KB)  TX bytes:39283 (39.2 KB)
```

- Configure the network discovery script to use this interface to query the network for BACnet devices (configuration stored in BACpypes.ini file).

```
[BACpypes]
objectName: Betelgeuse
address: 10.0.2.15/24:47808
objectIdentifier: 599
maxApduLengthAccepted: 1024
segmentationSupported: segmentedBoth
vendorIdentifier: 15
```

- Helpful link for Subnet Mask: [https://www.aelius.com/njh/subnet\\_sheet.html](https://www.aelius.com/njh/subnet_sheet.html)



# Running the Network Discovery Script

► To Run the utility execute the following command:

- `python bacnet_scan.py`
  - The output should be similar to this:

```
Device Address = <Address 192.168.1.42>
Device Id      = 699
maxAPDULengthAccepted = 1024
segmentationSupported = segmentedBoth
vendorID       = 15

Device Address = <RemoteStation 1002:11>
Device Id      = 540011
maxAPDULengthAccepted = 480
segmentationSupported = segmentedBoth
vendorID       = 5
```

- The critical parameters are the “Device Address” and “Device ID”.
  - These parameters are necessary for configuration of drivers for data trending.
- The vendorID can be helpful for identification of a specific device.
  - <http://www.bacnet.org/VendorID/BACnet%20Vendor%20IDs.htm>

# Generating a BACnet Registry Configuration File



Pacific Northwest  
NATIONAL LABORATORY

Proudly Operated by **Battelle** Since 1965

- ▶ A registry configuration file is needed to allow the VOLTTRON™ driver to scrape points off of a device.
- ▶ The registry file can be generated using the `grab_bacnet_config.py` script. To Run the utility execute the following command:
  - `python grab_bacnet_config.py`
    - Note: Previous to VOLTTRON 3.5 `grab_bacnet_config.py` took the device address as an argument instead of the device ID.
  - This will query the device with the matching device ID for configuration information and print the resulting output to the console.
  - In order to save the configuration to a file use the `--out-file` option to specify the output file name.
  - Optionally the `--address` option can be used to specify the address of the target. In some cases, this is needed to help establish a route to the device.

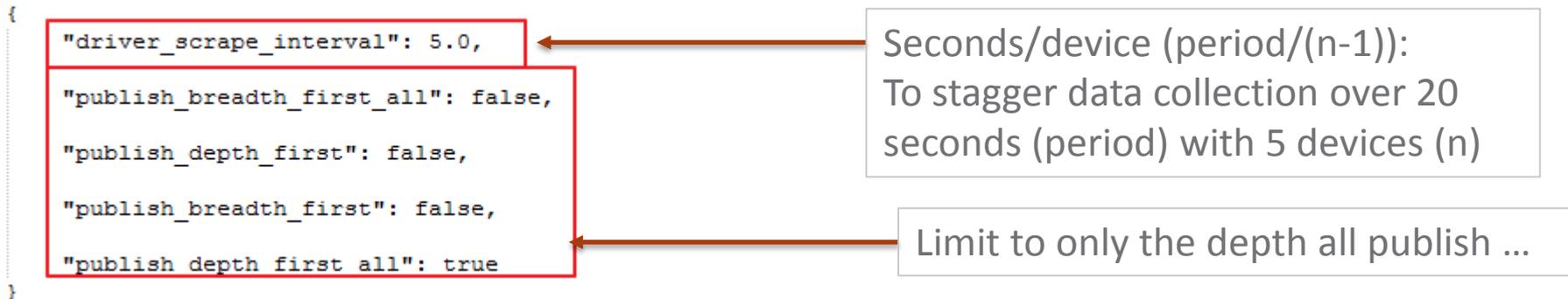
# Edit the BACnet Registry Configuration File

- ▶ Modify the VOLTTRON™ Point Name to the desired trending name (i.e., the name published to the VOLTTRON™ message bus and inserted into database by Historian):
  - By default, the “**Voltron Point Name**” is set to the value of the name property of the BACnet object (point) on the device. A duplicate of “**Voltron Point Name**” column called “**Reference Point Name**” is created so that once “**Voltron Point Name**” is changed a reference remains to the actual BACnet device object name.
  
- ▶ Modify the Writeable field:
  - Attempts at determining if a point is writable proved too unreliable. Therefore all points are considered to be read-only in the output.
  - Change this field to “TRUE” if the point is known to be writeable.
    - AnalogOutput/BinaryOutput – Almost always writeable
    - AnalogValue/BinaryValue – Usually writeable
    - AnalogInput/BinaryInput – Almost never writeable
  - BACnet priority for writeable points maybe specified (default write priority is 16, lowest priority).



# MasterDriverAgent Configuration

- ▶ The MasterDriverAgent requires a configuration file:



- ▶ [http://voltron.readthedocs.io/en/develop/core\\_services/drivers/Driver-Configuration.html#master-driver-agent-configuration](http://voltron.readthedocs.io/en/develop/core_services/drivers/Driver-Configuration.html#master-driver-agent-configuration)



# Driver Configuration

- ▶ Each trended device requires a configuration file:

```
{  
  "driver_config": {"device_address": "20:0x000000000000",  
                  "device_id": 0},  
  "driver_type": "bacnet",  
  "registry_config": "config://registry_configs/ahu.csv",  
  "interval": 60,  
  "timezone": "UTC"  
}
```

From BACnet scan (bacnet\_scan.py)

CSV file from grab\_bacnet\_config.py

- ▶ For a list of additional driver configuration parameters:
  - [http://voltron.readthedocs.io/en/master/core\\_services/drivers/Driver-Configuration.html](http://voltron.readthedocs.io/en/master/core_services/drivers/Driver-Configuration.html)



# Configure and Launch the BACnetProxy

- ▶ Communication with BACnet devices on a network happens via a single virtual BACnet device. In the new driver architecture, we have a separate agent specifically for communicating with BACnet devices and managing the virtual BACnet device.
- ▶ This agent uses the same configuration parameters used when modifying the BACpypes.ini file:

```
{  
  "agentid": "bacnet_proxy",  
  
  #Maximum APDU length accepted  
  #This setting determines the largest APDU accepted by the Volttron BACnet virtual device.  
  #Valid options are 50, 128, 206, 480, 1024 (default), and 1476  
  #"max_apdu_length": 1024,  
  
  #ID of the Device object of the virtual bacnet device.  
  #Defaults to 599  
  "object_id": 100,  
  
  #Name of the bacnet network object  
  #Defaults to "Volttron BACnet driver"  
  #"object_name": "Volttron BACnet driver",  
  
  #Vendor ID of the virtual bacnet device.  
  #Defaults to 15  
  #"vendor_id": 15,  
  
  #Required, use this network interface for the virtual device.  
  "device_address": "10.0.2.15/24:47808"  
}
```

- ▶ [http://voltron.readthedocs.io/en/master/core\\_services/drivers/BACnet-Proxy-Agent.html](http://voltron.readthedocs.io/en/master/core_services/drivers/BACnet-Proxy-Agent.html)

# Launching the MasterDriverAgent and BACnetProxy

- ▶ “make” scripts can be used to easily build/rebuild, install, and start an agent with one command.
- ▶ A template for creating “make” files is located at:
  - [volttron-directory]/scripts/core/make-listener
    - This “make”, when executed will build/rebuild, install, and start the ListenerAgent. This script can be copied and modified to build any agent.

# Start Device Data Trending – Adding Device Configurations to the VOLTRON™ Configuration Store



Pacific Northwest  
NATIONAL LABORATORY

Proudly Operated by **Battelle** Since 1965

- ▶ The configuration store provides storage for agent configurations and an agent interface to facilitate dynamic agent configuration.
- ▶ [http://voltron.readthedocs.io/en/master/core\\_services/config\\_store/index.html](http://voltron.readthedocs.io/en/master/core_services/config_store/index.html)
- ▶ After adding the device and registry configurations to the configuration store the MasterDriverAgent will scrape data off the devices and publish the data to the message bus.



# Device Driver Configuration

## ▶ Questions/Discussion