



NIAGARA SUMMIT 2026

SEAMLESS CONNECTIVITY,
POWERFUL INTELLIGENCE

TRIDIUM 

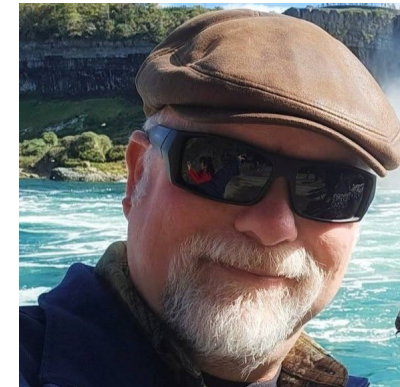
Mastering Niagara: Hidden Gems, Tips and Troubleshooting



NIAGARA
SUMMIT
2026



Jerome Bergquist



James Johnson



Eric Anderson



Kyle Sardinia

This document is a non-binding, confidential document that contains valuable proprietary and confidential information of Tridium and must not be disclosed to any third party without our written agreement. It does not create any binding obligations on us to develop or sell any product, service or offering. Content provided herein cannot be altered or modified and must remain in the format as originally presented by Tridium. All demos included in this presentation are for informational purposes only.

Agenda

- Common Troubleshooting Techniques
- Remote programming and subscription
- BACnet tuning tips
- Bog file protection
- Offline user management

Troubleshooting



Troubleshooting a Station

- Engine Watchdog Timeout
- Runtime and Memory
- Spy Views
- How to read application director
- What information should you grab from the station?

Engine Watchdog Timeout

- The Niagara application uses an Engine Watchdog Timer thread that totalizes time.
- The station's main engine thread resets the watchdog timer at the end of each scan.
- If the watchdog timer exceeds the timeout value, that indicates the main engine thread is stuck on a scan, and the engine watchdog policy is enforced.
- The default policy is Terminate to restore station functionality.
- Symptoms of engine watchdog timeouts include unexpected station restart, and lower station runtime than expected.
- These are confirmed by syslog records and backup console text file.

Platform Service Container Plugin

Locale	en
System Time	16:29
Date	30-Mar-2026
Time Zone	America/Chicago (-6/-5)
Engine Watchdog Policy	Terminate
Engine Watchdog Timeout	00000h 03m [0 ms - +inf]

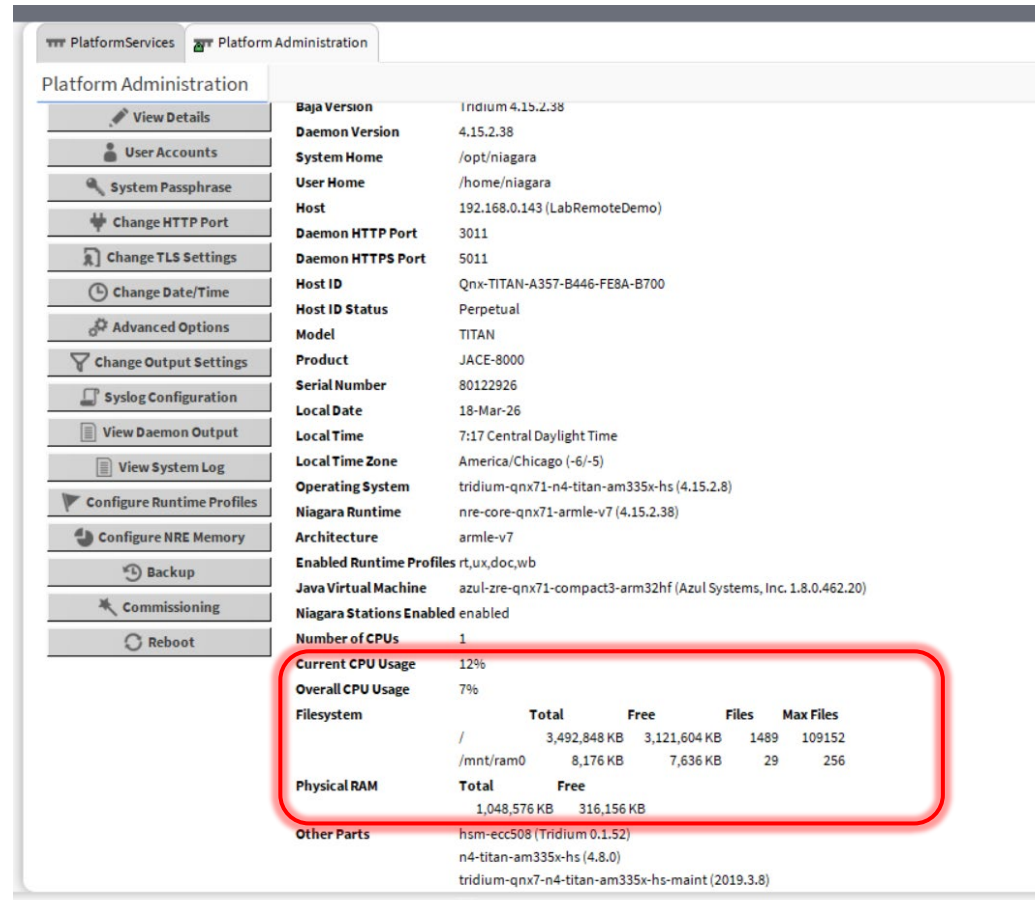
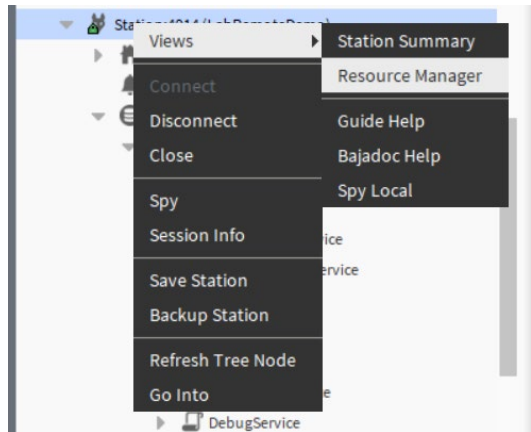
Engine Watchdog Policy	Terminate
Engine Watchdog Timeout	Log Only
Enable Station Auto-Save	Terminate
Station Auto-Save Frequency	Reboot

Runtime and Memory

Where to look:

- Resource Manager
- Platform Administration
- Platform Services

Resource Manager



Platform Administration

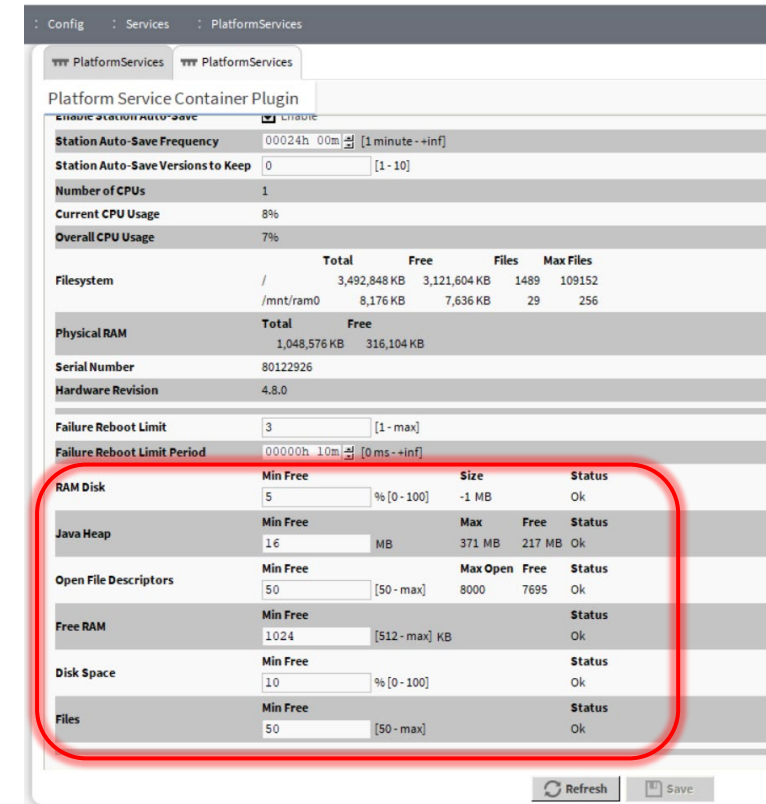
Baja Version Indium 4.15.2.38
Daemon Version 4.15.2.38
System Home /opt/niagara
User Home /home/niagara
Host 192.168.0.143 (LabRemoteDemo)
Daemon HTTP Port 3011
Daemon HTTPS Port 5011
Host ID Qnx-TITAN-A357-B446-FE8A-B700
Host ID Status Perpetual
Model TITAN
Product JACE-8000
Serial Number 80122926
Local Date 18-Mar-26
Local Time 7:17 Central Daylight Time
Local Time Zone America/Chicago (-6/-5)
Operating System tridium-qnx71-n4-titan-am335x-hs (4.15.2.8)
Niagara Runtime nre-core-qnx71-armle-v7 (4.15.2.38)
Architecture armle-v7
Enabled Runtime Profiles rt,ux,doc,wb
Java Virtual Machine azul-zre-qnx71-compact3-arm32hf (Azul Systems, Inc. 1.8.0.462.20)
Niagara Stations Enabled enabled
Number of CPUs 1
Current CPU Usage 12%
Overall CPU Usage 7%

Filesystem	Total	Free	Files	Max Files
/	3,492,848 KB	3,121,604 KB	1489	109152
/mnt/ram0	8,176 KB	7,636 KB	29	256

Physical RAM	Total	Free
	1,048,576 KB	316,156 KB

Other Parts
hsm-ecc508 (Tridium 0.1.52)
n4-titan-am335x-hs (4.8.0)
tridium-qnx71-n4-titan-am335x-hs-maint (2019.3.8)

Platform Administration
Should be no more than ~<75% on average



Platform Service Container Plugin

Station Auto-Save Frequency 00024h 00m [1 minute - +inf]
Station Auto-Save Versions to Keep 0 [1 - 10]
Number of CPUs 1
Current CPU Usage 8%
Overall CPU Usage 7%

Filesystem	Total	Free	Files	Max Files
/	3,492,848 KB	3,121,604 KB	1489	109152
/mnt/ram0	8,176 KB	7,636 KB	29	256

Physical RAM	Total	Free
	1,048,576 KB	316,104 KB

Serial Number 80122926
Hardware Revision 4.8.0

Failure Reboot Limit 3 [1 - max]
Failure Reboot Limit Period 00000h 10m [0 ms - +inf]

RAM Disk	Min Free	Size	Status
	5	% [0 - 100]	-1 MB Ok

Java Heap	Min Free	Max	Free	Status
	16	MB	371 MB	217 MB Ok

Open File Descriptors	Min Free	Max Open	Free	Status
	50	[50 - max]	8000	7695 Ok

Free RAM	Min Free	Status
	1024	[512 - max] KB Ok

Disk Space	Min Free	Status
	10	% [0 - 100] Ok

Files	Min Free	Status
	50	[50 - max] Ok

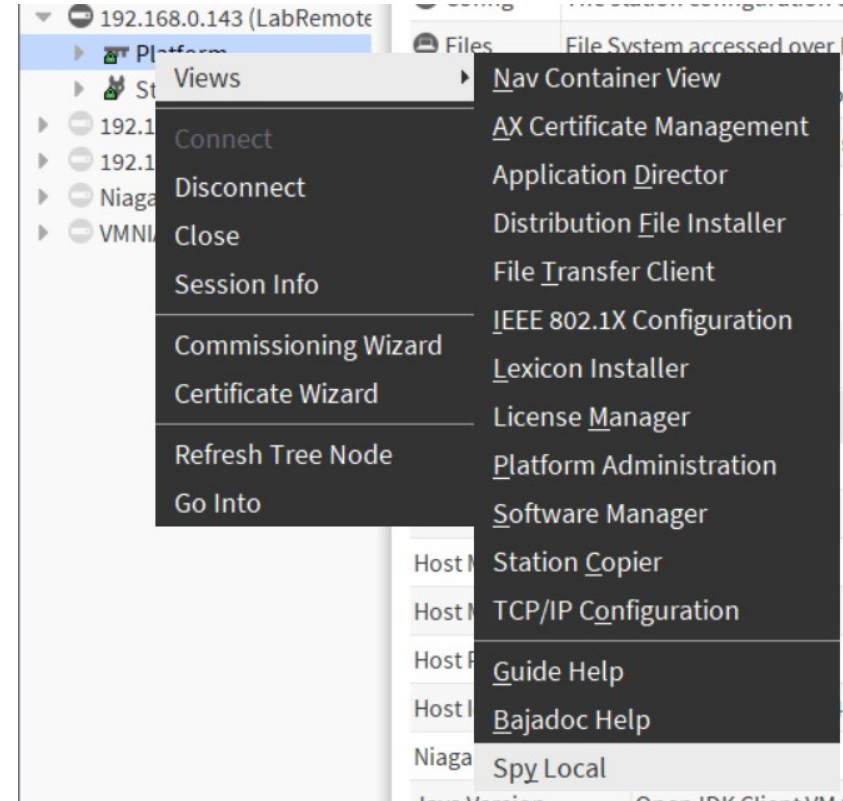
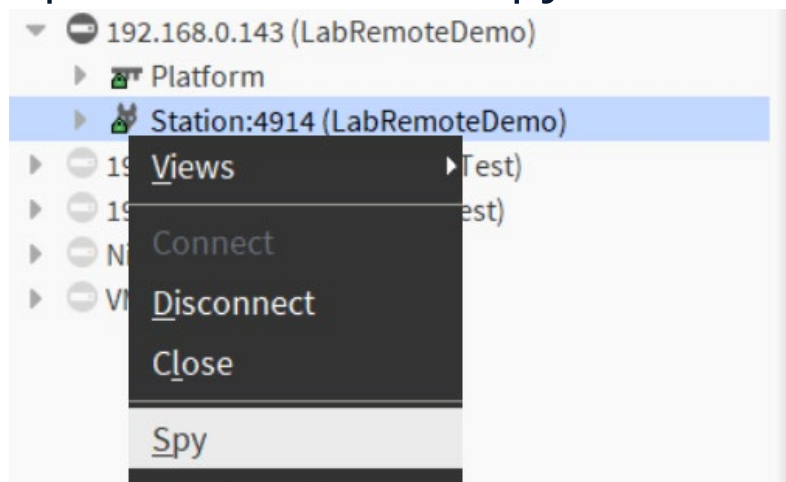
Platform Services
Checking for memory usage.

Checking the Spy Views

Local and Remote Spies, what is the difference?

- Remote Spy – Is the remote station (server)
- Local Spy – Is the Workbench proxy (client)

When you right click on the Station, Spy in the menu opens the Remote Spy.



When you right click on the Platform, under Views Spy Local will be at the bottom of the menu.

Spy engineManager

One of the most useful portions of the Spy Remote is the Engine Manager.

```
Remote Station | sysManagers | engineManager
```

atSteadyState	true
suspended	false
scanCount	25884119
runtime	520479656ms [6 days, 34 minutes, 39.656 seconds]
totalTicksInsideScan	2117152ms [35 minutes, 17.152 seconds]
averageTime/scan	0.08179347344215192ms
averageTimeLast10Scan	0.2ms

```
Peak Scan Stats (time in scan)
```

engine scan peak[0]	2205ms [2.205 seconds] @ 18 Mar 26 10:24:28.544 AM CDT
---------------------	--

- Select sysManagers → engineManager
- Includes details like the average scan times and other performance statistics.

- sysInfo
- stdout
- console
- systemProperties
- **sysManagers**
- util
- classLoaders
- metrics
- logSetup
- securityInfo
- nav
- fox
- platform diagnostics
- localHistory
- stationJetty
- niagaraNetwork
- Firewall Rules
- webFileCache
- webDevSetup
- hxPxCache
- workbench
- pxIncludeManager
- microImageManager
- registryManager
- schemaManager
- licenseManager
- moduleManager
- **engineManager**
- leaseManager
- serviceManager
- stationManager
- resourceManager
- diagnosticManager
- queryHandlers
- networkInterfaceManager
- scheduleManager
- dataRecoveryManager
- alarmRecordDecorators

Spy engineManager – Engine Hogs

Actions

currentComponent null

currentAction null

Async ActionQueue 0 (Peak 36)

Engine Hogs

Reset Engine Hogs

Reset Peak Scan Time

Remote Station | sysManagers | engineManager | hogs

Engine Hogs					
Rank	Component	Type	Total Time	Total Count	Avg time
0	/Sandbox/NumericWritable	control:NumericWritable	281596ms [4 minutes, 41.596 seconds]	520887	540 us
1	/Sandbox/SineWave	kitControl:SineWave	144875ms [2 minutes, 24.875 seconds]	520884	278 us
2	/Drivers/BacnetNetwork/MainAHU/points/W1	control:BooleanWritable	24835ms [24.835 seconds]	57236	433 us
3	/Drivers/BacnetNetwork/Rm\$201\$20Stat/points/SpaceTemp	control:NumericPoint	21254ms [21.254 seconds]	53642	396 us
4	/Drivers/BacnetNetwork/MainAHU/points/Space\$20Temperature	control:NumericPoint	21241ms [21.241 seconds]	52368	405 us
5	/Drivers/BacnetNetwork/MainAHU/points/Y1	control:BooleanWritable	6038ms [6.038 seconds]	52368	115 us
6	/Drivers/BacnetNetwork/MainAHU/points/System\$20Mode	control:EnumPoint	5406ms [5.406 seconds]	52367	103 us
7	/Drivers/BacnetNetwork/Rm\$201\$20Stat/points/Humidity	control:NumericPoint	2918ms [2.918 seconds]	53641	54 us
8	/Drivers/BacnetNetwork/Rm\$201\$20Stat/points/Max\$20Set\$20Temperature\$20Cooling	control:NumericWritable	2893ms [2.893 seconds]	10387	278 us
9	/Drivers/BacnetNetwork/Rm\$201\$20Stat/points/Max\$20Set\$20Temperature\$20Heating	control:NumericWritable	2511ms [2.511 seconds]	10388	241 us
10	/Drivers/BacnetNetwork/MainAHU/points/Active\$20Cool\$20Setpoint	control:NumericPoint	2440ms [2.440 seconds]	52367	46 us
11	/Drivers/BacnetNetwork/Rm\$201\$20Stat/points/Fan	control:BooleanPoint	2245ms [2.245 seconds]	53640	41 us
12	/Drivers/BacnetNetwork/Rm\$201\$20Stat/points/Desired\$20Set\$20Temperature\$20Heating	control:NumericWritable	2243ms [2.243 seconds]	10407	215 us
13	/Drivers/BacnetNetwork/MainAHU/points/Tstat	kitControl:Tstat	2243ms [2.243 seconds]	20623	108 us
14	/Drivers/BacnetNetwork/MainAHU/points/Active\$20Heat\$20Setpoint	control:NumericPoint	1868ms [1.868 seconds]	52367	35 us
15	/Drivers/BacnetNetwork/Rm\$201\$20Stat/points/Illuminance	control:NumericPoint	1852ms [1.852 seconds]	53640	34 us
16	/Drivers/BacnetNetwork/MainAHU/points/G	control:BooleanWritable	1744ms [1.744 seconds]	4870	358 us
17	/Drivers/BacnetNetwork/MainAHU/points/Tstat1	kitControl:Tstat	1387ms [1.387 seconds]	20623	67 us
18	/Drivers/BacnetNetwork/Rm\$201\$20Stat/points/ExternalOccupancyOverride	control:BooleanWritable	1315ms [1.315 seconds]	10390	126 us
19	/Drivers/BacnetNetwork/Rm\$201\$20Stat/points/Desired\$20Set\$20Temperature\$20Cooling	control:NumericWritable	1095ms [1.095 seconds]	10406	105 us
20	/Drivers/BacnetNetwork/Rm\$201\$20Stat/points/Min\$20Set\$20Temperature\$20Cooling	control:NumericWritable	1090ms [1.090 seconds]	10387	105 us
21	/Drivers/BacnetNetwork/Rm\$201\$20Stat/points/Min\$20Set\$20Temperature\$20Heating	control:NumericWritable	1084ms [1.084 seconds]	10388	104 us
22	/Drivers/BacnetNetwork/Rm\$201\$20Stat/points/Active\$20Set\$20Temperature\$20Heating	control:NumericPoint	1009ms [1.009 seconds]	10387	97 us
23	/Drivers/BacnetNetwork/Rm\$201\$20Stat/points/Active\$20Conditioning\$20Mode	control:EnumPoint	920ms [920 ms]	10405	88 us
24	/Drivers/BacnetNetwork/Rm\$201\$20Stat/points/Temperature\$20Offset	control:NumericWritable	897ms [897 ms]	10387	86 us
25	/Services/SecurityService/certificates/tridium/expiry	nss:CertificateExpiryPoint	402ms [402 ms]	148	2720 us

The Engine Hogs view will display the top 100, listed 0-99, along with the details of each.

Spy platform diagnostics

Platform diagnostics in Remote Spy

- Select platform diagnostics
- Then select either “View CPU usage by process” or “View CPU usage by thread”
- Note: this is also where to collect a diagnostics file your support channel may request for further help.

- sysInfo
 - stdout
 - console
 - systemProperties
 - sysManagers
 - util
 - classLoaders
 - metrics
 - logSetup
 - securityInfo
 - nav
 - fox
 - **platform diagnostics**
 - localHistory
 - stationJetty
 - niagaraNetwork
 - Firewall Rules
 - webFileCache
 - webDevSetup
 - hxPxCache
 - workbench
 - pxIncludeManager
 - microlmageManager
- Niagara Station JMX Info
- View current date status
 - View current uptime
 - View network hostname
 - View network domainname
 - View network interfaces
 - View network routes
 - View network dns status
 - View socket statistics
 - View process table
 - View CPU usage by process**
 - View CPU usage by thread**
 - View memory usage
 - Report free flash space
 - Report free uSD space
 - Report Niagara disk usage
 - View firewall rules
 - Start log dropped packets
 - View log dropped packets
 - Stop log dropped packets
 - View system log
 - Collect diagnostics file**

Using the Application Director

- Combing through the Application Director can tell you a lot about what is happening in your station.
- Clicking the Pause Output button pauses the stream, making it easier to read.
- When the output is paused, clicking the Load Output button resumes the stream.
- Stream To File will create a .txt file that you can review in another application or send along for support.

The screenshot displays the Application Director interface. At the top, it shows the application name 'LabRemoteDemo' in a 'station' type, which is currently 'Running'. The details include 'fox=n/a,foxs=4914,foxwss=443,http=n/a,https=443', 'Auto-Start' set to 'true', and 'Restart on Failure' set to 'true'. Below this is a log window showing various system messages, including 'Launching Niagara Runtime Environment', 'Booting', 'Logging initialized', and 'Niagara runtime booted'. On the right side, there is a control panel with several buttons: 'Start', 'Stop', 'Restart', 'Reboot', 'Kill', 'Dump Threads', 'Save Bog', 'Verify Software', 'Clear Output', 'Pause Output', 'Output Dialog', 'Stream To File', and 'Output Settings'. The 'Pause Output' button is highlighted with a red box, and the 'Stream To File' button is highlighted with a blue box.

Using the Application Director

- The exception details should be read from the bottom to the top within the stack trace.
- Each line defines the java class and method being called in the code.
- The numbers at the end of each line reference which line of code is being executed within the referenced object.

```
SEVERE [09:48:01 04-Apr-23 EDT][niagaraDriver] Error receiving msg: /Drivers/NiagaraNetwork/VykonProJace/schedules
javax.baja.naming.UnresolvedException: /GlobalSchedules/GymSchedule
  at javax.baja.naming.BSlotScheme.doResolve(BSlotScheme.java:289)
  at javax.baja.naming.BSlotScheme.resolve(BSlotScheme.java:95)
  at javax.baja.naming.BOrdScheme.resolve(BOrdScheme.java:124)
  at javax.baja.naming.BOrd.resolve(BOrd.java:290)
  at javax.baja.naming.BOrd.resolve(BOrd.java:254)
  at javax.baja.naming.BOrd.resolve(BOrd.java:234)
  at com.tridium.nd.schedule.BScheduleChannel.process(BScheduleChannel.java:138)
  at com.tridium.fox.sys.BFoxConnection.process(BFoxConnection.java:470)
  at com.tridium.fox.session.SessionDispatcher.dispatch(SessionDispatcher.java:92)
  at com.tridium.fox.session.SessionDispatcher.run(SessionDispatcher.java:68)
  at java.lang.Thread.run(Thread.java:750)
```

Watchdog Timeout

```
ENGINE WATCHDOG TIMEOUT STACK DUMP @ Tue Mar 31 10:05:00 EDT 20262026-03-31 10:05:00
Full thread dump OpenJDK 64-Bit Server VM (25.472-b08 mixed mode):
```

```
"Nre:Engine" #14 daemon prio=5 os_prio=0 tid=0x0000026ee67f6800 nid=0x1fa4 waiting on condition [0x000000f7b1eff000]
  java.lang.Thread.State: TIMED_WAITING (sleeping)
  at java.lang.Thread.sleep(Native Method)
  at Prog_e8de2b39b3a64457acca7eb0d4f06d70.onExecute (Prog_e8de2b39b3a64457acca7eb0d4f06d70.java:27)
  at com.tridium.program.BProgram.doExecute (BProgram.java:152)
  at auto.com.tridium_program_BProgram.invoke (AutoGenerated)
  at com.tridium.sys.schema.ComponentSlotMap.invoke (ComponentSlotMap.java:1921)
  at com.tridium.sys.engine.EngineUtil.doInvoke (EngineUtil.java:62)
  at com.tridium.sys.engine.EngineManager.checkAsyncActions (EngineManager.java:471)
  at com.tridium.sys.engine.EngineManager.execute (EngineManager.java:272)
  at com.tridium.sys.engine.EngineManager$EngineThread.run (EngineManager.java:911)
```

```
23 public void onExecute() throws Exception
24 {
25     boolean test = true;
26     while(test) {
27         Thread.sleep(5000L);
28     }
29 }
```

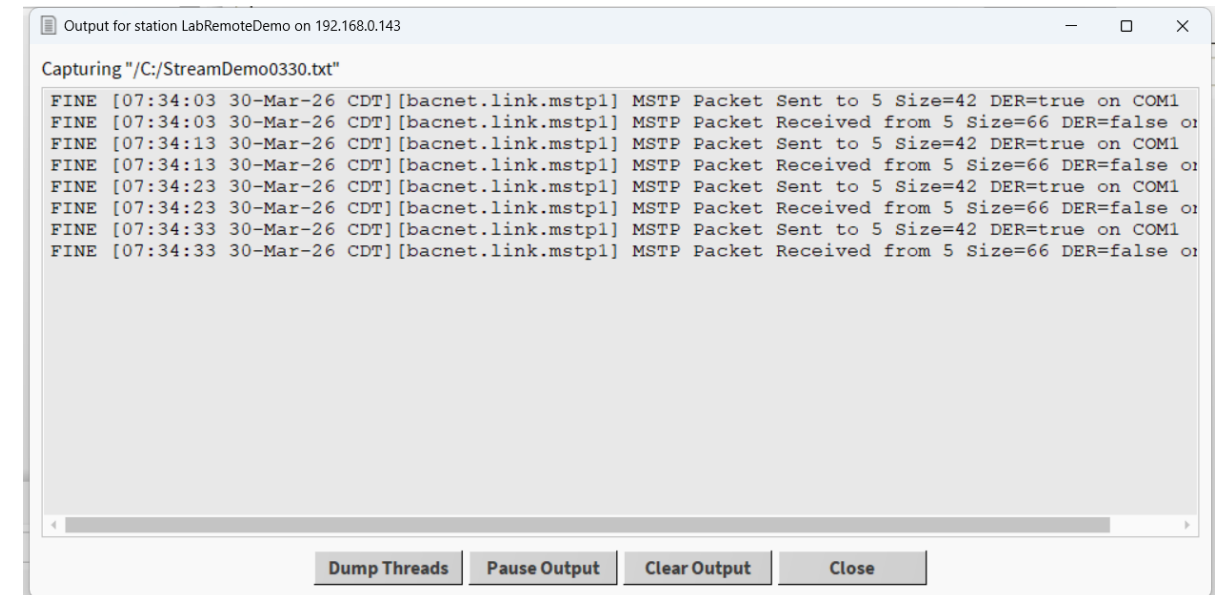
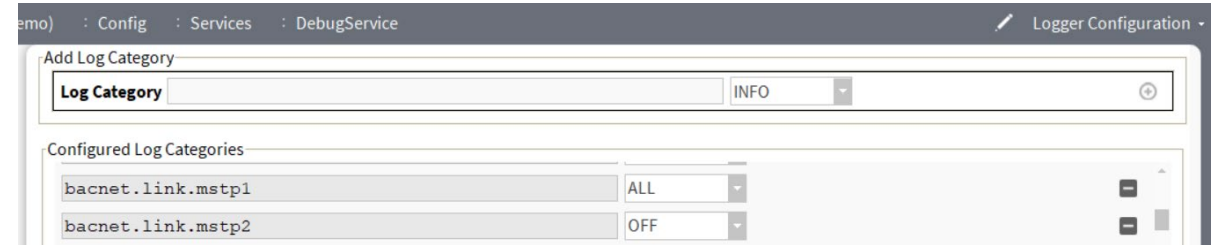
- Watchdog timeout event means the main engine thread is blocked
- Review the Nre:Engine thread stack trace
- Read from the bottom-up
- The line number in parenthesis indicates the specific line of source code being executed

Program2 (Program)	
Code Program Code	
Class Name	<u>Prog_e8de2b39b3a64457acca7eb0d4f06d70</u>
Class File	0xcafefabeb00000034001c
Dependencies	program-rt;nre;baja

Using the Debug Service

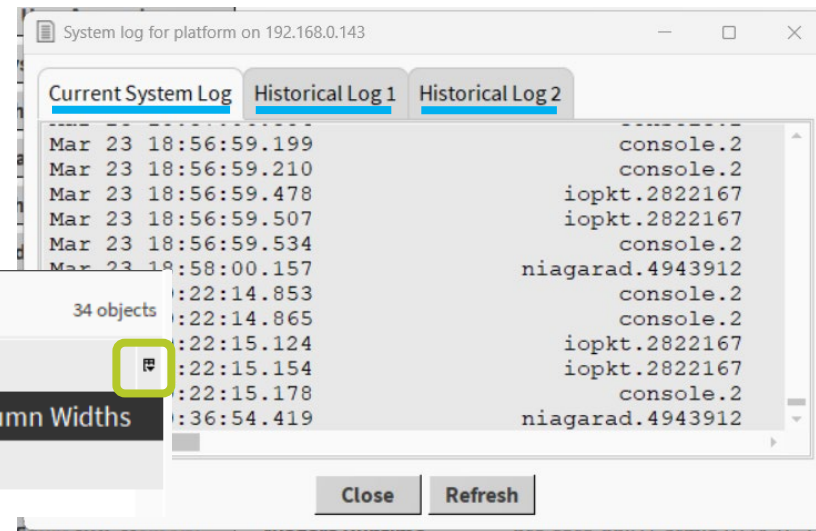
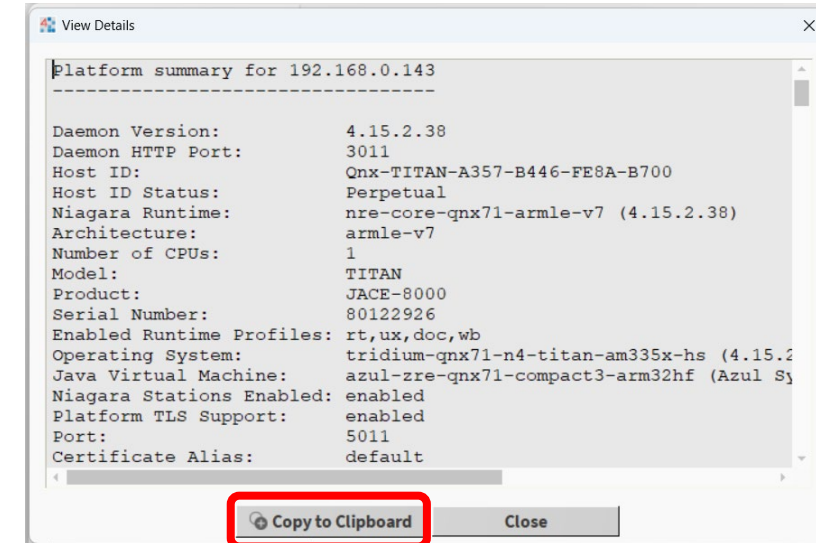
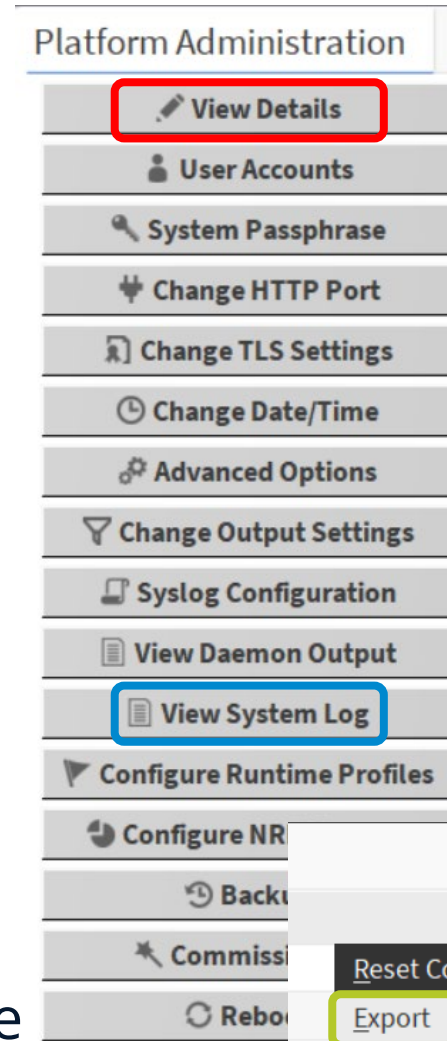
To change the verbosity of logs printing in the Application Director Stream

- Open DebugService
- Find a log category you're looking for detail on.
- Set to a more verbose level, or all, then save the Logger Configuration view.
- Return log settings to the default info level when finished debugging.



What happens when you're stumped?

- What should you include for Technical Support to best help you.
 - From Platform Administration:
 - Station version and hardware details
 - View Details
 - Copy to Clipboard
 - Save to .txt file.
 - System Logs
 - View System Log
 - Copy and paste the 3 logs to .txt files
 - Application Director output:
 - The stream file that was produced from the Application Director including exceptions
 - Consider dumping threads multiple times
 - Diagnostic File from the Platform Spy
 - Export of the Resource Manager Table



Remote Programming and Component Subscription

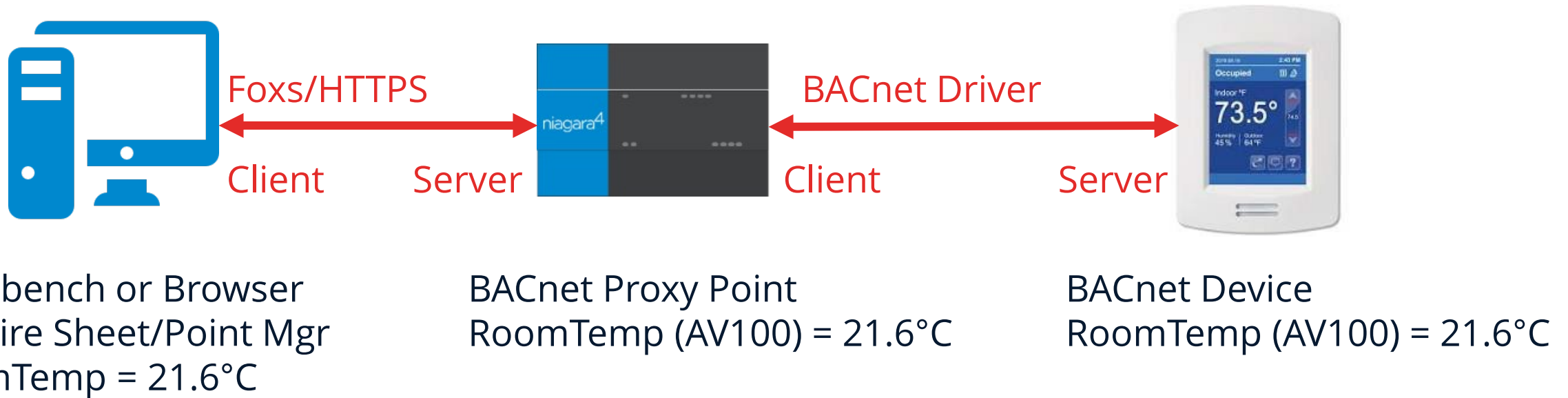


Remote Programming - Overview

- Broadly applies to using the component model across a network connection.
- Applies to both client Workbench or browser connected to a station, and a driver connected to a field device.
- Is one of the most powerful features of Niagara.
- Is a common cause of confusion and performance problems.

Proxy

- Mechanism which allows the client or driver to model a component from the server (station or field device).
- Can maintain the state of the server component by synchronizing properties.



Proxy States

- **Unloaded** – the proxy has not been loaded across the network.
- **Loaded** – the proxy is loaded across the network; it may or may not be out-of-date with the server/device.
- **Subscribed** – the proxy is actively synchronized with the server/device.

Component Subscription

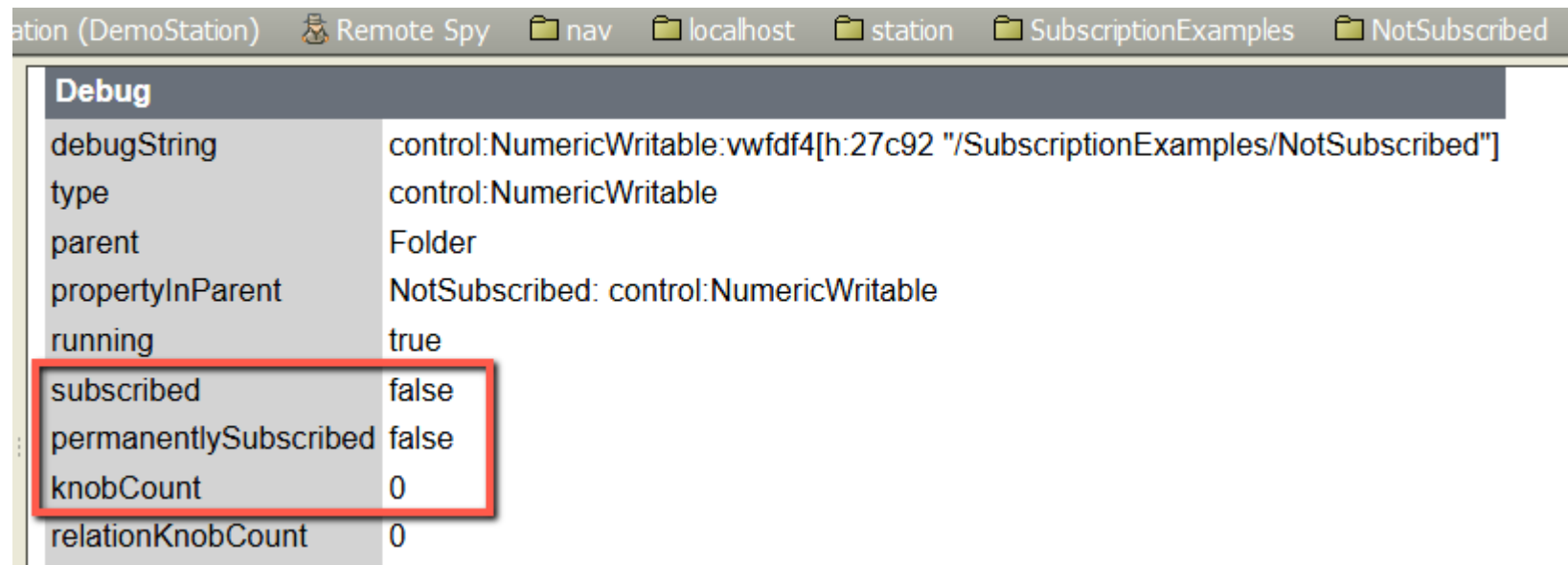
A component is subscribed when there is an active interest

- Using Workbench or browser client to view a property sheet, wire sheet, point manager, Px widgets, etc.
- Control point with an active history extension.
- Control point with an alarm extension.
- Control point with a control extension.
- Component is a source of a wire sheet link.
- Other components may lease or subscribe in their code.

Red = permanently subscribed

Component Subscription

- Spy views do not cause subscription, but right clicking on a component may cause transient subscription.
- Resolving a BQL query does not cause subscription, but station components or Px widgets that utilize BQL queries may cause subscription.



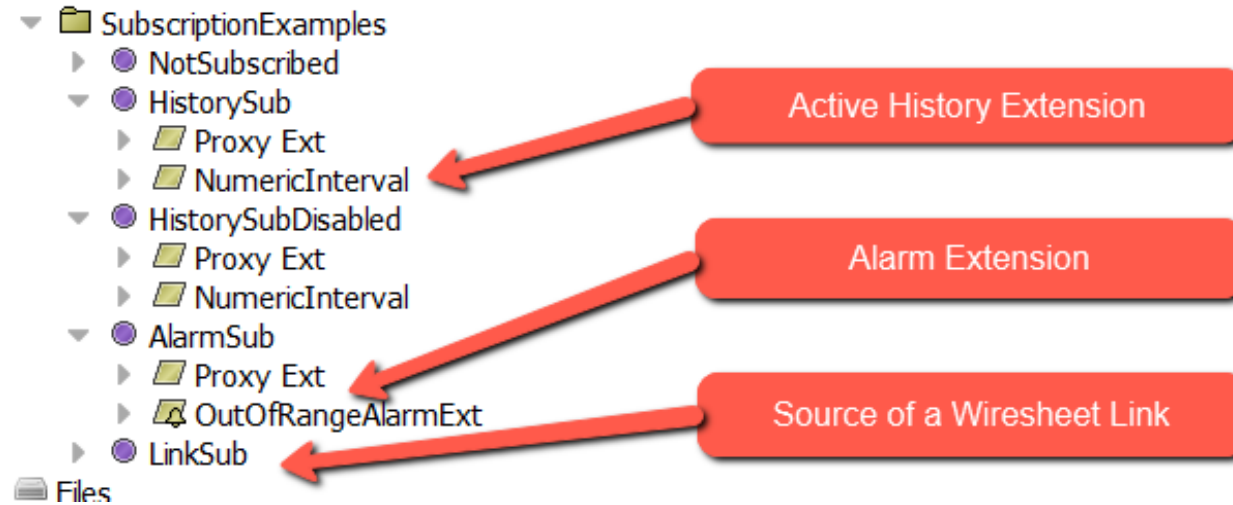
The screenshot shows a debug console window with a breadcrumb trail: station (DemoStation) > Remote Spy > nav > localhost > station > SubscriptionExamples > NotSubscribed. Below the breadcrumb is a table of component properties. The 'subscribed' property is highlighted with a red box and has a value of 'false'.

Debug	
debugString	control:NumericWritable:wvdf4[h:27c92 "/SubscriptionExamples/NotSubscribed"]
type	control:NumericWritable
parent	Folder
propertyInParent	NotSubscribed: control:NumericWritable
running	true
subscribed	false
permanentlySubscribed	false
knobCount	0
relationKnobCount	0

Component Subscription

The image shows five component panels, each representing a different subscription state:

- NotSubscribed:** Out: 67.6 °F {ok} @ 16; In10: - {null}; In16: 67.6 °F {ok}
- HistorySub:** Out: 67.6 °F {ok} @ 16; In10: - {null}; In16: 67.6 °F {ok}
- HistorySubDisabled:** Out: 67.6 °F {ok} @ 16; In10: - {null}; In16: 67.6 °F {ok}
- AlarmSub:** Out: 67.6 °F {unackedAlarm} @ 16; In10: - {null}; In16: 67.6 °F {ok}
- LinkSub:** Out: 67.6 °F {ok} @ 16; In10: - {null}; In16: 67.6 °F {ok}

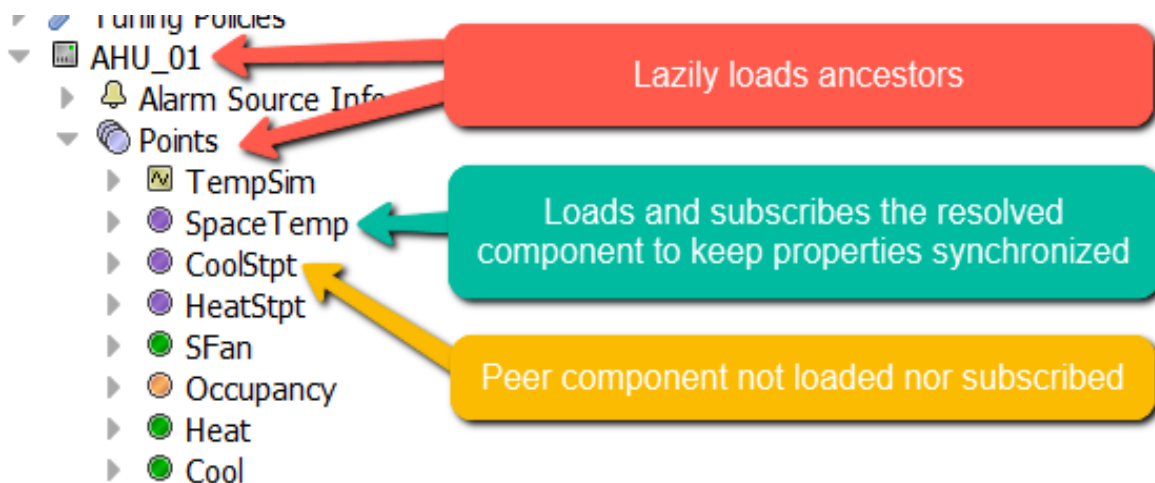


Name	Value With Facets	Is Subscribed	Is Permanently Subscribed
NotSubscribed	71.5 °F	false	false
HistorySub	71.5 °F	true	true
HistorySubDisabled	71.5 °F	false	false
AlarmSub	71.5 °F	true	true
LinkSub	71.5 °F	true	false

station:|slot:/SubscriptionExamples|bql:select name, valueWithFacets, isSubscribed, isPermanentlySubscribed from control:NumericWritable

Workbench & Browser Client Subscription

- Resolves a proxy of the server-side component and keeps properties synchronized.
- Proxy component inherits available actions.
- Ancestor components are lazily loaded.
- Loaded components are cached for duration of session.



	%out.value%	%parent.CoolStpt.out.value%
Space Temp	67.1 °F	Cool Stpt 0.0
▲ Bound Label Binding		
ord	station:/slot:/Drivers/BacnetNetwork/AHU_01/points/SpaceTemp	
degradeBehavior	None	
hyperlink	null	
summary	%displayName?typeDisplayName% = %.%	
popupEnabled	true	
statusEffect	None	

BQL and Bound Table – BACnet Example

Zone	Space Temp	Cool Stpt	Heat Stp	SFan	Occupancy	Heat	Cool
AHU_01	72.8 °F	73.0 °F	70.0 °F	On	Occupied	Off	Off
AHU_02	72.8 °F	73.0 °F	70.0 °F	On	Occupied	Off	Off
AHU_03	72.8 °F	73.0 °F	70.0 °F	On	Occupied	Off	Off
AHU_04	72.8 °F	73.0 °F	70.0 °F	On	Occupied	Off	Off
AHU_05	72.8 °F	73.0 °F	70.0 °F	On	Occupied	Off	Off
AHU_06	72.8 °F	73.0 °F	70.0 °F	On	Occupied	Off	Off
AHU_07	72.8 °F	73.0 °F	70.0 °F	On	Occupied	Off	Off
AHU_08	72.8 °F	73.0 °F	70.0 °F	On	Occupied	Off	Off
AHU_09	72.8 °F	73.0 °F	70.0 °F	On	Occupied	Off	Off
AHU_10	72.8 °F	73.0 °F	70.0 °F	On	Occupied	Off	Off

Bound Table does not subscribe components and does not update automatically

Following a station restart, transient slots such as the out slot on a control point are initialized with default values

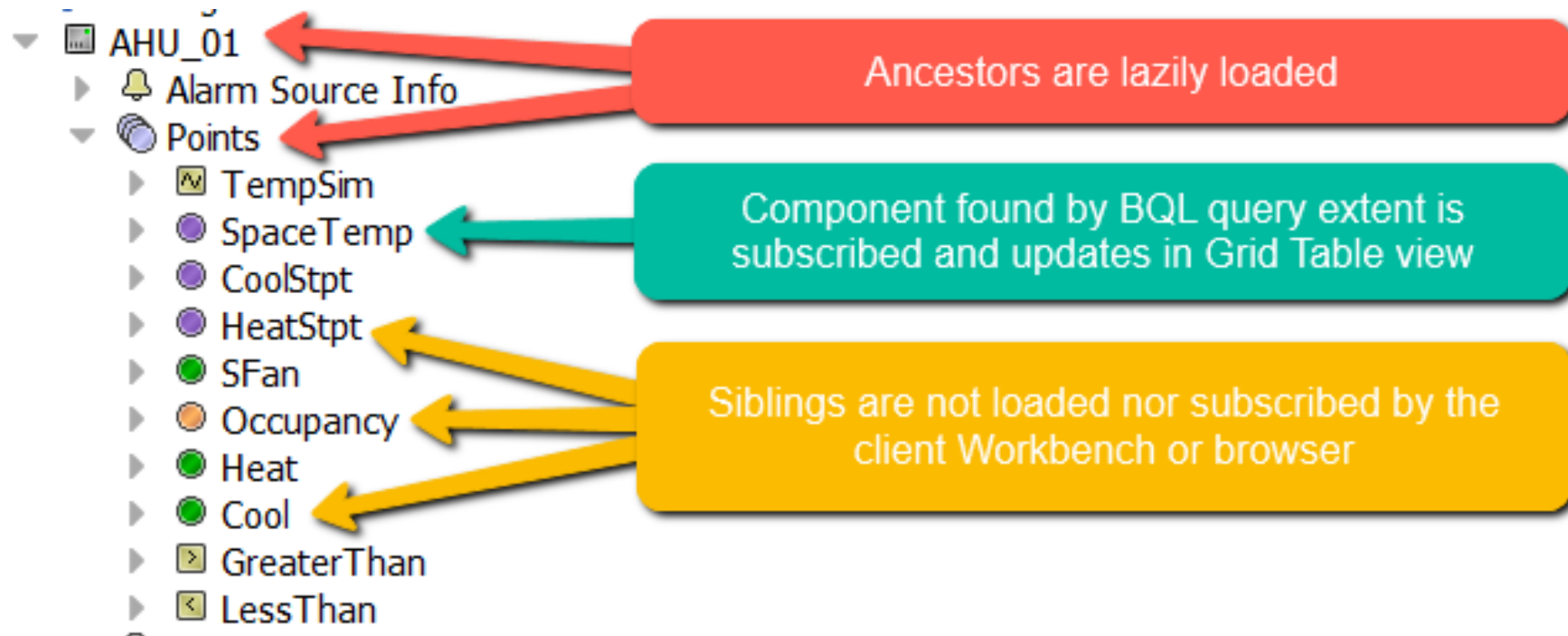
BQL query returns last known value for points

```
station:|slot:/Drivers/BacnetNetwork|bql:select parent.name as 'Zone', SpaceTemp.valueWithFacets as 'Space Temp', CoolStpt.valueWithFacets as 'Cool Stpt', HeatStpt.valueWithFacets as 'Heat Stp', SFan.valueWithFacets as 'SFan', Occupancy.valueWithFacets as 'Occupancy', Heat.valueWithFacets as 'Heat', Cool.valueWithFacets as 'Cool' from bacnet:BacnetPointDeviceExt
```

The BQL query is resolved server side in the station and the static results are sent to the client to display in the Bound Table widget.

BQL Grid

- Subscribes the components found by BQL query extent.
- Ancestor components are lazily loaded.
- Sibling and child components are not loaded.



BQL Grid – Client Example

Zone	Space Temp	Cool Stpt	Heat Stpt	SFan	Occupancy	Heat	Cool
AHU_01	67.1 °F	0.0	0.0	false	0	false	false
AHU_02	67.1 °F	0.0	0.0	false	0	false	false
AHU_03	67.1 °F	0.0	0.0	false	0	false	false
AHU_04	67.1 °F	0.0	0.0	false	0	false	false
AHU_05	67.1 °F	0.0	0.0	false	0	false	false
AHU_06	67.1 °F	0.0	0.0	false	0	false	false
AHU_07	67.1 °F	0.0	0.0	false	0	false	false
AHU_08	67.1 °F	0.0	0.0	false	0	false	false
AHU_09	67.1 °F	0.0	0.0	false	0	false	false
AHU_10	67.1 °F	0.0	0.0	false	0	false	false

```
station:|slot:/Drivers/BacnetNetwork|bql:select parent.parent.name as 'Zone', valueWithFacets as 'Space Temp', parent.CoolStpt.valueWithFacets as 'Cool Stpt', parent.HeatStpt.valueWithFacets as 'Heat Stpt', parent.SFan.valueWithFacets as 'SFan', parent.Occupancy.valueWithFacets as 'Occupancy', parent.Heat.valueWithFacets as 'Heat', parent.Cool.valueWithFacets as 'Cool' from control:NumericPoint where name = 'SpaceTemp'
```

On the client side, the Grid Table resolves the BQL query and receives an ord reference to the component found by the extent of the query for each row. In this example, the client subscribes only the SpaceTemp components.

Component Grid

- Resolves each cell to a component which is loaded and subscribed.

Zone slot: %displayName%	Space Temp slot:points/SpaceTemp %valueWithFacets%	Cool Stpt slot:points/CoolStpt %valueWithFacets%	Heat Stpt slot:points/HeatStpt %valueWithFacets%	SFan slot:points/SFan %valueWithFacets%	Occupancy slot:points/Occupancy %valueWithFacets%	Heat slot:points/Heat %valueWithFacets%	Cool slot:points/Cool %valueWithFacets%
--------------------------------	--	--	--	---	---	---	---

station:|slot:/Drivers/BacnetNetwork/AHU_01

 Empty Query

station:|slot:/Drivers/BacnetNetwork/AHU_01

station:|slot:/Drivers/BacnetNetwork/AHU_02

station:|slot:/Drivers/BacnetNetwork/AHU_03

station:|slot:/Drivers/BacnetNetwork/AHU_04

station:|slot:/Drivers/BacnetNetwork/AHU_05

station:|slot:/Drivers/BacnetNetwork/AHU_06

station:|slot:/Drivers/BacnetNetwork/AHU_07

station:|slot:/Drivers/BacnetNetwork/AHU_08

station:|slot:/Drivers/BacnetNetwork/AHU_09

station:|slot:/Drivers/BacnetNetwork/AHU_10

Component Grid – Client Example

Zone	Space Temp	Cool Stpt	Heat Stpt	SFan	Occupancy	Heat	Cool
AHU_01	69.6 °F	73.0 °F	70.0 °F	On	Occupied	On	Off
AHU_02	69.6 °F	73.0 °F	70.0 °F	On	Occupied	On	Off
AHU_03	69.6 °F	73.0 °F	70.0 °F	On	Occupied	On	Off
AHU_04	69.6 °F	73.0 °F	70.0 °F	On	Occupied	On	Off
AHU_05	69.6 °F	73.0 °F	70.0 °F	On	Occupied	On	Off
AHU_06	69.6 °F	73.0 °F	70.0 °F	On	Occupied	On	Off
AHU_07	69.6 °F	73.0 °F	70.0 °F	On	Occupied	On	Off
AHU_08	69.6 °F	73.0 °F	70.0 °F	On	Occupied	On	Off
AHU_09	69.6 °F	73.0 °F	70.0 °F	On	Occupied	On	Off
AHU_10	69.6 °F	73.0 °F	70.0 °F	On	Occupied	On	Off

On the client side, the Grid Table view resolves the relative column ord against the component for each row. This allows the client to resolve the ord to the component in each cell, and subscribes all components to display up to date values in each cell.

BACnet Polling



Poll List Entries Per Proxy Point

- Main entry for proxy point's property- usually Present Value
- Add Status Flags property if
 - "statusFlags" facet is true
 - Point is for the Present Value property
 - Object has a Status Flags property
- Add Event State if "eventState" facet is true
- Add Priority Array if "priorityArray" facet is true
- Add Reliability if "reliability" facet is true
- Add for any BacnetDeviceObjectPropertyReferences on proxy ext

Poll List Entries Per Proxy Point

- Status Flags decoded to BStatus
- Event State (state) and Reliability (reliability) added to facets
- Priority Array used to set “bac” facet to “def” or 1-16
- BacnetDeviceObjectPropertyReferences added to facets keyed by property ID

Subscribing to Points

1. Get network port poll service based on device network number
 2. Add poll list entries to dibs bucket (dibsSubTime)
 3. Get poll list entries (getPLEsSubTime)
 4. Increment point count
 5. Add poll list entries to bucket based on poll frequency (fast, normal, slow) (pleAddTime)
- subTime is duration of steps 2-5

Subscribing to Points

Remote Station | nav | localhost | station | Drivers | BacnetNetwork | bacnetComm | network | ipPort | pollService

subs	500
subTime	16
subAvg	0.03
unsubs	0
unsubTime	0
unsubAvg	nan
getPLEsSubTime	0
getPLEsSubAvg	0.00
pleAdds	500
pleAddTime	0
pleAddAvg	0.00
dibsSubTime	16
dibsSubAvg	0.03
getPLEsUnsubTime	0
getPLEsUnsubAvg	0.00
pleRems	0
pleRemTime	0
pleRemAvg	nan
dibsUnsubTime	0
dibsUnsubAvg	nan

Poll Lists

- Contains poll list entries for the same device
- Number of entries limited by smaller of local or remote Max APDU Length Accepted
- Entries grouped by object ID
- Fit poll list in unsegmented message
- Add “splitOnFailure” Integer property to poll service to split the poll list after that many failures (if > 0) (splitCount)
- Add “repackInterval” RelTime property to periodically rebuild (recombine) poll lists

Poll Lists

The screenshot shows the Niagara Workbench interface. On the left, a navigation tree shows the hierarchy: Platform > Station (bacnetPolling_01) > Config > Services > Drivers > NiagaraNetwork > BacnetNetwork > Local Device > Bacnet Comm > Network > Router Table > Ip Port > Link > Poll Service. The main window displays the configuration for the selected 'Poll Service'.

The configuration is shown in a table-like view with the following fields:

- norm.index: 28
- norm.pollCount: 28

The main area contains a list of poll lists, each with a 'norm' index and a detailed description of the poll list configuration. The descriptions include device information, PLEs, and a list of analog values and their present values.

```
norm[0] 1 PollList [bacnetPolling_02 device:2 1:127.0.0.1:47812] d9dad4c {91} -D [0] 5 PLEs analogValue:1 presentValue in bacnetPolling_02 [228437324] -> analogValue:1:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:2 presentValue in bacnetPolling_02 [228437324] -> analogValue:2:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:3 presentValue in bacnetPolling_02 [228437324] -> analogValue:3:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:4 presentValue in bacnetPolling_02 [228437324] -> analogValue:4:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:5 presentValue in bacnetPolling_02 [228437324] -> analogValue:5:Present Value:-1:REAL {11}; Bacnet.pointCx
```

```
norm[1] 1 PollList [bacnetPolling_02 device:2 1:127.0.0.1:47812] d9dad4c {91} -D [0] 5 PLEs analogValue:6 presentValue in bacnetPolling_02 [228437324] -> analogValue:6:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:7 presentValue in bacnetPolling_02 [228437324] -> analogValue:7:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:8 presentValue in bacnetPolling_02 [228437324] -> analogValue:8:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:9 presentValue in bacnetPolling_02 [228437324] -> analogValue:9:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:10 presentValue in bacnetPolling_02 [228437324] -> analogValue:10:Present Value:-1:REAL {11}; Bacnet.pointCx
```

```
norm[2] 1 PollList [bacnetPolling_02 device:2 1:127.0.0.1:47812] d9dad4c {91} -D [0] 5 PLEs analogValue:11 presentValue in bacnetPolling_02 [228437324] -> analogValue:11:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:12 presentValue in bacnetPolling_02 [228437324] -> analogValue:12:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:13 presentValue in bacnetPolling_02 [228437324] -> analogValue:13:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:14 presentValue in bacnetPolling_02 [228437324] -> analogValue:14:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:15 presentValue in bacnetPolling_02 [228437324] -> analogValue:15:Present Value:-1:REAL {11}; Bacnet.pointCx
```

```
norm[3] 1 PollList [bacnetPolling_02 device:2 1:127.0.0.1:47812] d9dad4c {91} -D [0] 5 PLEs analogValue:16 presentValue in bacnetPolling_02 [228437324] -> analogValue:16:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:17 presentValue in bacnetPolling_02 [228437324] -> analogValue:17:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:18 presentValue in bacnetPolling_02 [228437324] -> analogValue:18:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:19 presentValue in bacnetPolling_02 [228437324] -> analogValue:19:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:20 presentValue in bacnetPolling_02 [228437324] -> analogValue:20:Present Value:-1:REAL {11}; Bacnet.pointCx
```

```
norm[4] 1 PollList [bacnetPolling_02 device:2 1:127.0.0.1:47812] d9dad4c {91} -D [0] 5 PLEs analogValue:21 presentValue in bacnetPolling_02 [228437324] -> analogValue:21:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:22 presentValue in bacnetPolling_02 [228437324] -> analogValue:22:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:23 presentValue in bacnetPolling_02 [228437324] -> analogValue:23:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:24 presentValue in bacnetPolling_02 [228437324] -> analogValue:24:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:25 presentValue in bacnetPolling_02 [228437324] -> analogValue:25:Present Value:-1:REAL {11}; Bacnet.pointCx
```

```
norm[5] 1 PollList [bacnetPolling_02 device:2 1:127.0.0.1:47812] d9dad4c {91} -D [0] 5 PLEs analogValue:26 presentValue in bacnetPolling_02 [228437324] -> analogValue:26:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:27 presentValue in bacnetPolling_02 [228437324] -> analogValue:27:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:28 presentValue in bacnetPolling_02 [228437324] -> analogValue:28:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:29 presentValue in bacnetPolling_02 [228437324] -> analogValue:29:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:30 presentValue in bacnetPolling_02 [228437324] -> analogValue:30:Present Value:-1:REAL {11}; Bacnet.pointCx
```

```
norm[6] 1 PollList [bacnetPolling_02 device:2 1:127.0.0.1:47812] d9dad4c {91} -D [0] 5 PLEs analogValue:31 presentValue in bacnetPolling_02 [228437324] -> analogValue:31:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:32 presentValue in bacnetPolling_02 [228437324] -> analogValue:32:Present Value:-1:REAL {11}; Bacnet.pointCx analogValue:33 presentValue in bacnetPolling_02 [228437324] -> analogValue:33:Present Value:-1:REAL {11}; Bacnet.pointCx
```

Poll List Order

- Default is based on order of subscriptions
- Add subclass of BacnetPollOrder to the poll service
- Ordered within each poll frequency bucket
- BacnetBalancedPollOrder: spread poll lists over network numbers
- BacnetRandomPollOrder

BacnetMultiPoll Next

1. Get poll list from dibs up to Max Dibs Polls (hidden, def: 5);
increment dibsPollTotal
 2. Get unexecuted, idle poll list from overdue bucket,
 - If possible, whose device is not already being polled
 - Then, even if the device is being polled
 - If found, increment bucket.pollTotal
- When all poll lists in a bucket have been executed, increment bucketcycleTotal

Polling RPM

1. Construct Read Property Multiple request
2. Send request and wait for response
3. Process response
4. Increment totalPollCount
5. Add poll time (steps 1-3) to totalPollTime

Polling without RPM

- Device does not support RPM service
- Receive Reject reason Unrecognized Service
- Receive Reject reason Buffer Overflow and maxAPDU has been reduced to minimum (50)

Poll Statistics

- $\text{averagePoll} = \text{totalPollTime} / \text{totalPollCount}$
 - Does not account for number of poll list entries in each poll
 - Many short polls might have a lower average
- $\text{totalPollTimePerThread} = \text{totalPollTime} / \text{numberOfThreads}$
- $\text{busyTime} = \text{totalPollTimePerThread} / \text{uptime}$
 - “{busyTime}% ({totalPollTime}/{uptime} over {numberOfThreads} thds)”

Poll Statistics

- totalPolls = “{totalPollCount} over {totalPollTime}”
- dibsPolls, fastPolls, normalPolls, slowPolls:
 - bucketPollCount / totalPollCount
 - “{bucketTotal}% ({bucketPollCount}/{totalPollCount})”
- dibsCount, fastCount, normalCount, slowCount:
 - current = bucketSize
 - average = bucketSizeSum / statsCount
 - bucketSizeSum += bucketSize

Poll Statistics

- fastCycleTime, normalCycleTime, slowCycleTime:
 - average = uptime / bucketCycleTotal

Poll Statistics Example

- normal poll frequency: 5 s
- uptime = 1 hr = 3600 s
- averagePoll = 15 ms

Low Load, 1 Thread

- 100 normal poll lists
- $100 \times 15 \text{ ms per poll} = 1.5 \text{ s polling per cycle}$
- $\text{cycles} = 3600 \text{ s} / 5 \text{ s} = 720 \text{ cycles}$
- $\text{totalPollTime} = 720 \text{ cycles} \times 1.5 \text{ s} = 1080 \text{ s}$
- $\text{totalPollTimePerThread} = 1080 \text{ s} / 1 \text{ thread} = 1080 \text{ s}$
- $\text{busyTime} = 1080 \text{ s} / 3600 \text{ s} = 30\%$
- $\text{cycleTime} = 3600 \text{ s uptime} / 720 \text{ cycles} = 5 \text{ s}$

Low Load, 2 Threads

- 100 normal poll lists
- $100 \times 15 \text{ ms per poll} = 1.5 \text{ s polling per cycle}$
- $\text{cycles} = 3600 \text{ s} / 5 \text{ s} = 720 \text{ cycles}$
- $\text{totalPollTime} = 720 \text{ cycles} \times 1.5 \text{ s} = 1080 \text{ s}$
- $\text{totalPollTimePerThread} = 1080 \text{ s} / 2 \text{ threads} = 540 \text{ s}$
- $\text{busyTime} = 540 \text{ s} / 3600 \text{ s} = 15\%$
- $\text{cycleTime} = 3600 \text{ s uptime} / 720 \text{ cycles} = 5 \text{ s}$

High Load, 1 Thread

- 1000 normal poll lists
- $1000 \times 15 \text{ ms per poll} = 15 \text{ s}$ polling per cycle
- $\text{cycles} = 3600 \text{ s} / 15 \text{ s} = 240 \text{ cycles}$
- $\text{totalPollTime} = 240 \text{ cycles} \times 15 \text{ s} = 3600 \text{ s}$
- $\text{totalPollTimePerThread} = 3600 \text{ s} / 1 \text{ thread} = 3600 \text{ s}$
- $\text{busyTime} = 3600 \text{ s} / 3600 \text{ s} = 100\%$
- $\text{cycleTime} = 3600 \text{ s uptime} / 240 \text{ cycles} = 15 \text{ s}$

High Load, 5 Threads

- 1000 normal poll lists
- $1000 \times 15 \text{ ms per poll} = 15 \text{ s polling per cycle}$
- $\text{cycles} = 3600 \text{ s} / 5 \text{ s} = 720 \text{ cycles}$
- $\text{totalPollTime} = 720 \text{ cycles} \times 15 \text{ s} = 10,800 \text{ s}$
- $\text{totalPollTimePerThread} = 10,800 \text{ s} / 5 \text{ threads} = 2160 \text{ s}$
- $\text{busyTime} = 2160 \text{ s} / 3600 \text{ s} = 60\%$
- $\text{cycleTime} = 3600 \text{ s uptime} / 720 \text{ cycles} = 5 \text{ s}$

Number of Threads

- No more than 2 on an MS/TP port
- Could add more than 2 to other port types (IP, Ethernet, SC)
- Too many threads could slow things down- don't drive busyTime to zero
- Consider other adjustments- poll frequency, tuning policy, number of points

More Info

- docBacnet – Troubleshooting
- docEngNotes – BACnet and Niagara 4

Live Demo

Bog File Protection and Offline Users



Objectives:

- Protection Mechanisms
- 4.15 Changes
- Bog Files
- Key Source



Protection Mechanisms



NIAGARA
SUMMIT
2026



Encrypting – Reversible with the correct Key



Hashing – irreversible, you can't get the original data

Encrypting

- **reversibleEncodingKeySource="keyring"** indicates that the encryption is sourced from the **NRE security provider's key ring**
- This is the default setting for **stations located in the daemon's** user home/stations directory
- The Station.bootStation() process **will fail** if the config.bog file does not encode passwords using this method
- The platform daemon (niagarad) automatically translates this to **external** when sending BOG files during backups or station transfers
- Only **authorized** calls can retrieve the key material

Encrypted hashed passwords

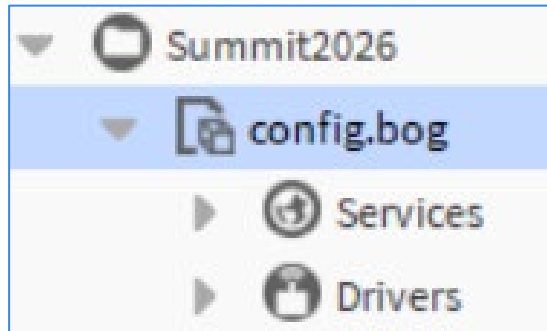
- As of Niagara 4.15, passwords are encrypted **in addition to** being hashed.
- A station with encrypted hashed passwords **will not run** in Niagara 4.14 or earlier. To downgrade the station, you must revert the station using the BOG File Protection tool.
- Station user passwords are encrypted and hashed on all platforms.

Bog Files

```
<?xml version="1.0" encoding="UTF-8"?>
<bajaObjectGraph version="4.0" reversibleEncodingKeySource="none" FIPSEnabled="false" reversibleEncodingValidator="[null.1]" >
<p h="2" c="1" m="b=baja" t="b:Station">
<p n="stationName" v="Summit2026"/>
<p n="Services" h="3" t="b:ServiceContainer">
<!-- /Services/AlarmService -->
<p n="AlarmService" h="4" m="a=alarm" t="a:AlarmService">
<p n="alarmDbConfig" h="5" t="a:FileAlarmDbConfig">
</p>
<p n="defaultAlarmClass" h="6" t="a:AlarmClass">
</p>
<p n="escalationTimeTrigger" h="7" m="c=control" t="c:TimeTrigger">
<p n="triggerMode" t="c:IntervalTriggerMode" v="false;00:00:00.000;23:59:59.999;60000;7f"/>
<p n="lastTrigger" t="b:AbsTime" v="2015-02-27T14:05:07.964-05:00"/>
<p n="nextTrigger" t="b:AbsTime" v="2015-02-27T14:06:07.956-05:00"/>
</p>
<a n="escalateAlarms" f="haAL"/>
</p>
<!-- /Services/BackupService -->
<p n="BackupService" h="8" m="bk=backup" t="bk:BackupService">
</p>
<!-- /Services/CategoryService -->
<p n="CategoryService" h="9" c="2" t="b:CategoryService">
<p n="ordMap" t="b:OrdToCategoryMap" v="history:^securityhistory=2&#xa;history:^audithistory=2&#xa;history:^loghistory=2&#xa;f
<!-- /Services/CategoryService/User -->
<p n="User" h="a" t="b:Category">
<p n="index" v="1"/>
</p>
<!-- /Services/CategoryService/Admin -->
<p n="Admin" h="b" t="b:Category">
<p n="index" v="2"/>
</p>
</p>
<!-- /Services/JobService -->
<p n="JobService" h="c" t="b:JobService">
</p>
<!-- /Services/SecurityService -->
<p n="SecurityService" h="697e" c="2" m="nss=nss" t="nss:SecurityService">
<p n="certificates" h="697f" t="nss:CertificateFolder">
</p>
</p>
<!-- /Services/RoleService -->
<p n="RoleService" h="d" c="2" t="b:RoleService">
<p n="admin" h="e" t="b:AdminRole">
<!-- /Services/RoleService/admin/hierarchy_RoleHierarchies -->
<p n="hierarchy_RoleHierarchies" h="f" m="hierarchy=hierarchy" t="hierarchy:RoleHierarchies">
</p>
```

- Niagara provides a standard XML format to store a tree of BValues. This XML format is called “BOG” for Baja Object Graph.
- Compact XML, using single letter element and attribute names
- Ability to compress using zip

Bog Files



- Bog files can be flat XML files or stored inside zip files. If zipped, then the zip file contains a single entry called “file.xml” with the XML document.
- You can use Workbench to copy any **BValue** to a directory on your file system to easily generate a bog file.
- Bog files are typically given a “.bog” extension. Bog files designed for use as palletes use the “.palette” extension, and though they use the same format as “.bog” files, they may not use reversible encodings for **BPassword** values.

Editing Offline .bog

The screenshot shows the 'Property Sheet' for 'config.bog (Station)'. The 'Station Name' is 'Summit2026'. Under the 'Services' section, there are several service containers listed: AlarmService, BackupService, CategoryService, JobService, SecurityService, RoleService, and UserService. The 'UserService' container is expanded, showing settings for 'Lock Out Enabled' (true), 'Lock Out Period' (+00000h 00m 10s), 'Max Bad Logins Before Lock Out' (5), 'Lock Out Window' (00000h 00m 30s), 'Default Auto Logoff Period' (00000h 15m), and 'Absolute Logoff Enabled' (true). Other settings include 'Default Absolute Logoff Period' (007d 00h 00m), 'User Prototypes', 'SMA Notification Settings', and 'License Expiration Notification Settings'.

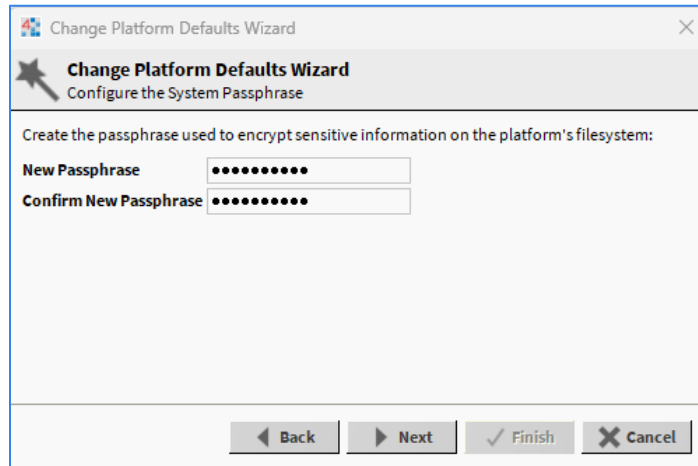
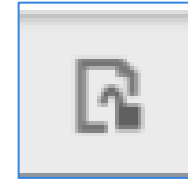
Property	Value
Station Name	Summit2026
Sys Info	>> ⌚
Services	Service Container
AlarmService	Alarm Service
BackupService	Backup Service
CategoryService	Category Service
JobService	Job Service
SecurityService	Security Service
RoleService	Role Service
UserService	User Service
Lock Out Enabled	<input checked="" type="checkbox"/> true
Lock Out Period	+00000h 00m 10s
Max Bad Logins Before Lock Out	5 [1 - 10]
Lock Out Window	00000h 00m 30s [1 second - 1 day]
Default Auto Logoff Period	00000h 15m [2 minutes - 4 hours]
Absolute Logoff Enabled	<input checked="" type="checkbox"/> true
Default Absolute Logoff Period	007d 00h 00m [15 minutes - 365 days]
User Prototypes	User Prototypes
SMA Notification Settings	SMA Notification Settings
License Expiration Notification Settings	License Expiration Notification Settings

- Files created in Workbench initially have no passphrase since they do not yet contain sensitive data.
- You can edit non-sensitive data without unlocking bog file protection using the passphrase

Editing Offline .bog (4.15)

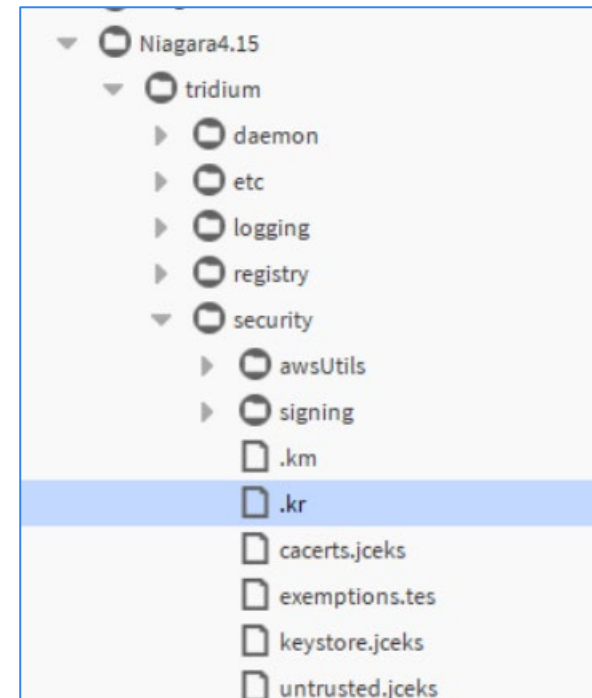
- In Niagara 4.15, you can add users by editing the station offline and creating the new user via the User Manager, or editing the .bog file directly.
- Because the encryption key is not yet available, the user password will be either only hashed or in plain text, depending on how the user was added.
- Once the station starts, the password will be hashed if needed and then encrypted.

Bog File Protection



- Bog files in a backup distribution file or under your Niagara user home are encrypted using a passphrase.

- Bog files in the Niagara daemon user home are encrypted using the host's key ring file.
- A unique key ring file is generated for each Niagara installation on a given host and is locked to that host.



Key Source

Key Source	Description	File Security and Portability
none	The file does not and cannot have any reversibly-encoded password values.	Best portability; consumers don't have to know a passphrase or have access to a particular system's "key ring".
keyring	Reversibly-encoded password values are encrypted using a randomly-generated secret key stored in a "key ring" on the system where the file resides.	Best security, but values are not readable on other systems.
external	Reversibly-encoded password values are encrypted using a secret key derived from a human-provided passphrase, which may be a controller's "system passphrase", may be assigned by Workbench user working with files offline, <i>etc.</i>	Portable. Correct passphrase needs to be supplied when reading encoded password values.

Questions

