Technical Document

Niagara$^{\text{AX-3.x}}$ Andover AC256 Driver Guide

Updated: March 19, 2008

Niagara$^{AX\text{-}3.x}$ Andover AC256 Driver Guide

Copyright © 2008 Tridium, Inc.
All rights reserved.

3951 Westerre Pkwy, Suite 350
Richmond
Virginia
23233
U.S.A.

Copyright Notice

The software described herein is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior written consent from Tridium, Inc.

The confidential information contained in this document is provided solely for use by Tridium employees, licensees, and system owners; and is not to be released to, or reproduced for, anyone else; neither is it to be used for reproduction of this Control System or any of its components.

All rights to revise designs described herein are reserved. While every effort has been made to assure the accuracy of this document, Tridium shall not be held responsible for damages, including consequential damages, arising from the application of the information contained herein. Information and specifications published here are current as of the date of this publication and are subject to change without notice.

The release and technology contained herein may be protected by one or more U.S. patents, foreign patents, or pending applications.

Trademark Notices

BACnet and ASHRAE are registered trademarks of American Society of Heating, Refrigerating and Air-Conditioning Engineers. Microsoft and Windows are registered trademarks, and Windows NT, Windows 2000, Windows XP Professional, and Internet Explorer are trademarks of Microsoft Corporation. Java and other Java-based names are trademarks of Sun Microsystems Inc. and refer to Sun's family of Java-branded technologies. Mozilla and Firefox are trademarks of the Mozilla Foundation. Echelon, LON, LonMark, LonTalk, and LonWorks are registered trademarks of Echelon Corporation. Tridium, JACE, Niagara Framework, Niagara^{AX} and Vykon are registered trademarks, and Workbench, WorkPlace^{AX}, and ^{AX}Supervisor, are trademarks of Tridium Inc. All other product names and services mentioned in this publication that is known to be trademarks, registered trademarks, or service marks are the property of their respective owners. The software described herein is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Niagara$^{AX-3.x}$ Andover AC256 Driver Guide

19 March 2008

This documents usage of the Andover AC256 driver for the NiagaraAX-3.1[1] (and later) framework.

[1] This driver includes changes made in the following NiagaraAX patch versions: 3.1.30.1, 3.2.16.1, and 3.3.1.1.

Compatibility

Tested Versions

The Andover AC256 driver was developed against the following AC256 systems:

1) AC256, Firmware version 8.3
   a. 1 Package Control Unit, Firmware version 3.3

The Andover AC256 driver has been field tested against the following AC256 systems:

1) AC256, Firmware version 7.0D
   a. 3 IOU panels
   b. 3 LCU panels

Platforms

The ANDOVER AC256 driver will function on all NiagaraAX platforms that support serial RS-232 communications. NiagaraAX-3.1 or higher is required.

Mapping of AC256 Points to Proxy Extensions

The Andover AC256 driver allows the creation of any of the following ControlPoint/ AndoverProxyExt combinations:

- NumericPoint with AndoverNumericProxyExt
- NumericWritable with AndoverNumericProxyExt
- BooleanPoint with AndoverBooleanProxyExt
- BooleanWritable with AndoverBooleanProxyExt
- StringPoint with AndoverStringProxyExt
- StringWritable with AndoverStringProxyExt
- EnumPoint with AndoverEnumProxyExt
- EnumWritable with AndoverEnumProxyExt

There are no limitations put on mapping AC256 points to proxy types. It is up to the integrator to determine which point type to create for each point that is learned from the Andover network.

To determine how point values obtained from the AC256 panels are converted to data for each of the above proxy point types, see "Point Value Conversion Considerations."

Installation

To use the NiagaraAX Andover AC256 driver, you must have a target host that is licensed for the feature “andoverAC256”. In addition, other device limits or proxy point limits may exist in your license.

From your PC, use the NiagaraAX Workbench 3*.1.x or later* installed with the “installation tool” option (checkbox “This instance of Workbench will be used as an installation tool”). This option installs the needed distribution files (*.dist* files) for commissioning various models of remote JACE platforms. The dist files are located under your Niagara install folder in various revision-named subfolders under the “sw” folder.

When installing Workbench on your PC, you should also select the **andoverAC256** module.

Apart from installing the 3*.n.nn* version of the NiagaraAX distribution files in the JACE, make sure to install the **andoverAC256** module too (if not already present, or upgrade if an older revision). For more details, see “About the Commissioning Wizard” in the JACE NiagaraAX Install and Startup Guide.

Following this, the station is now ready for ANDOVER AC256 software integration, as described in the rest of this document.

Quick Start

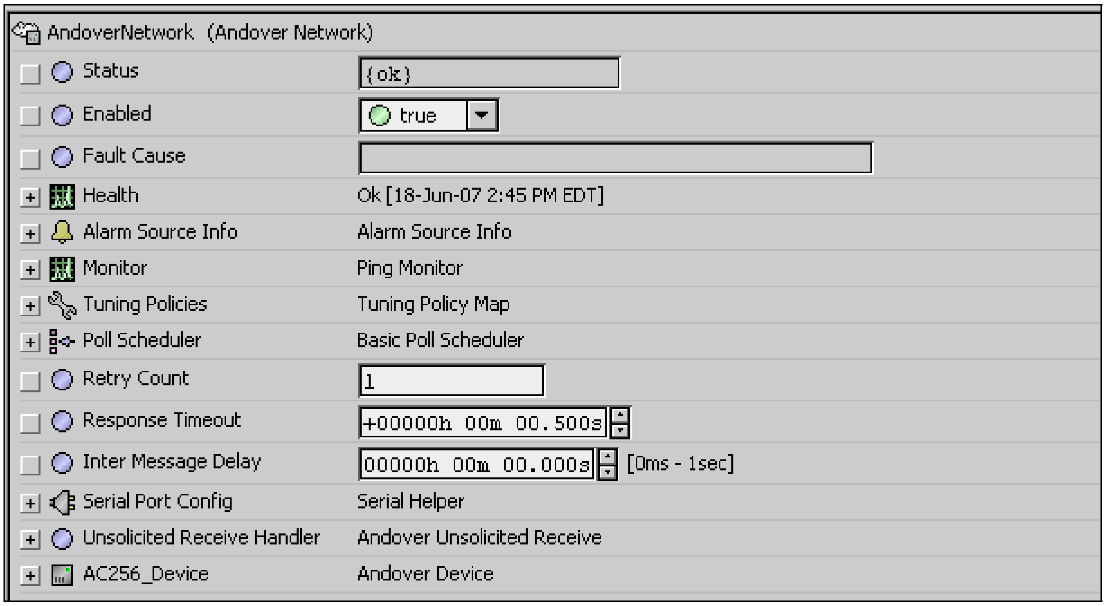
1. Create A new station
    - Follow the installation and configuration instructions preceding this.
    - On WorkBench menu “Tools”, select “New Station”.
    - In the “Nav” pane of WorkBench, expand the new station, and double-click the “drivers” branch. This will display a “Driver Manager” in the right pane.
    - Click “New” in the “Driver Manager”. Select the “Andover Network”, and click “Add”. Answer the remaining wizard prompts until the new network is added.
    - Right click on “config.bog” for the new station in the Nav tree, and select “Save”.
    - At this point you will need to download the new station to the JACE and start the station.
2. Download and Start the new Station on the target JACE.
3. Configure and Learn
    - Open the running station in Workbench
    - View the property sheet for the AndoverNetwork:
        - Response Timeout: 0.5 sec.
        - Retry Count: 1
        - Inter Message delay: 0.0 sec
        - Expand “Serial Port Parameters” and enter in the communication settings that match the setup of the AC256 panel you are connecting to.
    - View the Device Manager view of the AndoverNetwork. Click “New” to add a single Andover Device. Go to the new Andover Device property sheet.
        - Set the “Device Password” to “&xxxxx” type of password
        - Set the “Built in User Password” to the administrator level password from the factory
    - Save the station and restart the JACE.
    - Navigate to the “Points” folder under the device and double click the points folder to display the “Andover Point Manager” view.
        - Click the “Discover” button and wait for discovery to complete
        - Select one or more discovered points in the top pane and click “Add”
        - You now have one or more Andover points added, and they should be polling.

AC256 Component Guides

AndoverNetwork

The AndoverNetwork provides all the configuration parameters necessary to allow the driver to communicate with a network of a single Andover AC256 device + LCUs.

The AndoverNetwork is the "network-level" component in the NiagaraAX architecture. It has the standard network component properties such as status and enabled (see "Driver Architecture / Common network components" in the *NiagaraAX-3.x Drivers Guide* for more information), as well as properties unique to configuring an AC256 network.



AndoverNetwork (Andover Network)

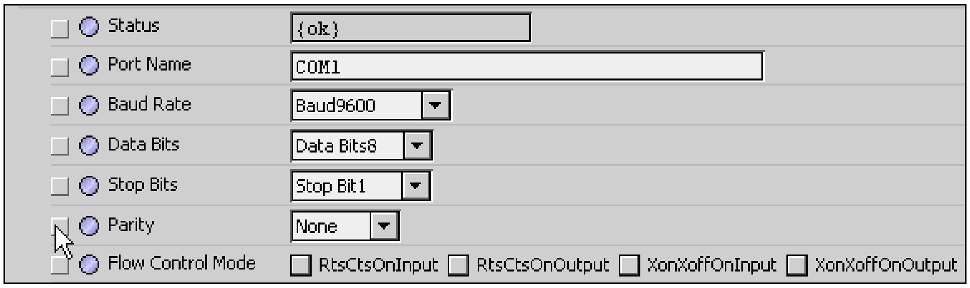
Status	{ok}
Enabled	true
Fault Cause	
Health	Ok [18-Jun-07 2:45 PM EDT]
Alarm Source Info	Alarm Source Info
Monitor	Ping Monitor
Tuning Policies	Tuning Policy Map
Poll Scheduler	Basic Poll Scheduler
Retry Count	1
Response Timeout	+00000h 00m 00.500s
Inter Message Delay	00000h 00m 00.000s [0ms - 1sec]
Serial Port Config	Serial Helper
Unsolicited Receive Handler	Andover Unsolicited Receive
AC256_Device	Andover Device

NOTE: In the following properties, the grayed out properties are inherited from the base NiagaraAX driver classes, and as such are only touched on here. For a full explanation, refer to the "Driver architecture" chapter in the NiagaraAX-3.x Drivers Guide.

- Status – The status of the network. Will normally be {ok}. See Fault Cause property for more information if not {ok}.
- Enabled – Enables or Disables the Andover AC256 Driver
- Fault Cause – if the "Status" is fault, the fault cause will appear here.
- Health – Communications health and fail statistics
- Alarm Source Info – configuration for routing alarms
- Monitor – container for monitor (ping) properties.
- Tuning Policies - A container for tuning policies which determines how and when proxy points are read and written.
- **Poll Scheduler –** contains properties for Fast, Normal, and Slow rate polling, as well as statistics on polling performance.
- **Retry Count –** Number of times to re-send a message if the initial try has no response.
- **Response Timeout –** The amount of time to wait for a response to a message sent to the AC256 panel. If this time is exceeded, then a retry will be sent (if "Retry Counts" is non-zero)

- **Inter Message Delay –** Some installations require a “pause” between adjacent messages to allow time for the AC256 panel to “catch up” with its own internal processing. This has the effect of adding time between polls, so this number should be kept as small as possible. Normally, a setting of “0.0” will work.

Serial Port Config – The communications parameters are set here to match the set up of the AC256 network controller’s RS-232 port.

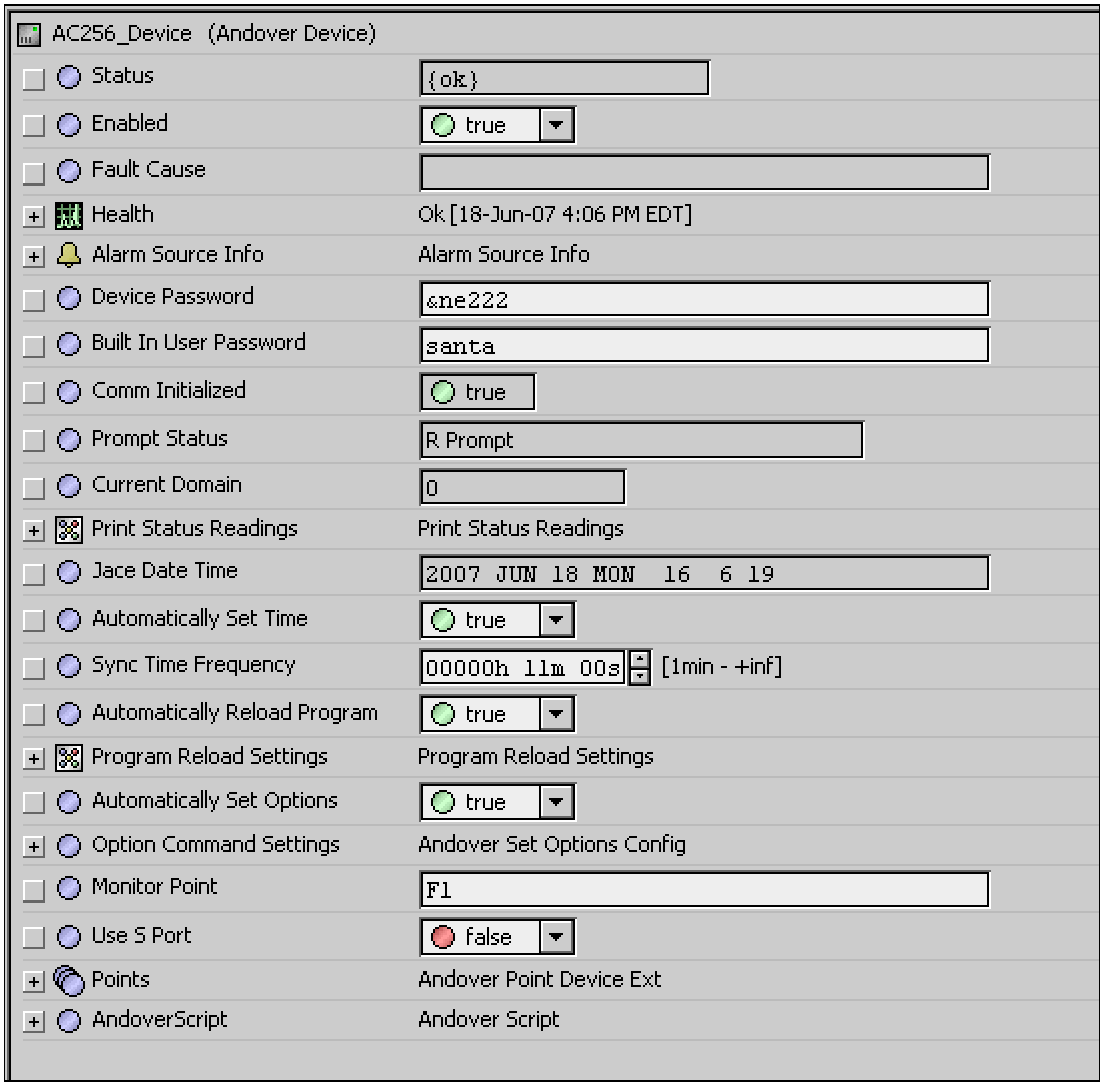


Status	{ok}
Port Name	COM1
Baud Rate	Baud9600
Data Bits	Data Bits8
Stop Bits	Stop Bit1
Parity	None
Flow Control Mode	☐ RtsCtsOnInput ☐ RtsCtsOnOutput ☐ XonXoffOnInput ☐ XonXoffOnOutput

- **Unsolicited Receive Handler –** no configurable or viewable properties.
- **Andover Device –** A dynamically added device that is a proxy for the actual AC256 panel. ***Note that an Andover Network should contain one and only one Andover Device***

AndoverDevice

The AndoverDevice is a dynamic slot on the AC256 Network. There should only ever be one and only one AndoverDevice located under an Andover Network object. The AndoverDevice corresponds to the AC256 panel to which the RS-232 connection is made.



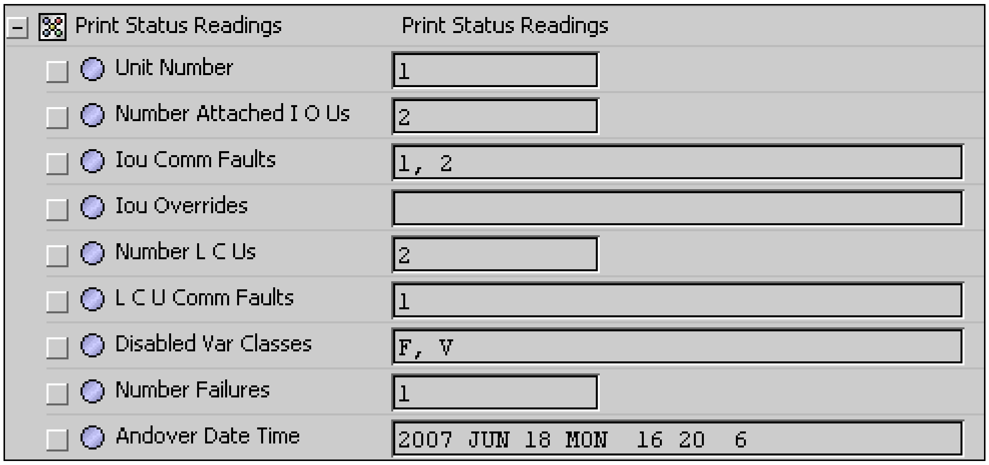
AC256_Device (Andover Device)

Status	{ok}
Enabled	true
Fault Cause	
Health	Ok [18-Jun-07 4:06 PM EDT]
Alarm Source Info	Alarm Source Info
Device Password	&ne222
Built In User Password	santa
Comm Initialized	true
Prompt Status	R Prompt
Current Domain	0
Print Status Readings	Print Status Readings
Jace Date Time	2007 JUN 18 MON 16 6 19
Automatically Set Time	true
Sync Time Frequency	00000h 11m 00s [1min - +inf]
Automatically Reload Program	true
Program Reload Settings	Program Reload Settings
Automatically Set Options	true
Option Command Settings	Andover Set Options Config
Monitor Point	F1
Use S Port	false
Points	Andover Point Device Ext
AndoverScript	Andover Script

NOTE: In the following properties, the grayed out properties are inherited from the base NiagaraAX driver classes, and as such are only touched on here. For a full explanation, refer to the "Driver architecture" chapter in the NiagaraAX-3.x Drivers Guide.

- Status – The status of the device. Will normally be {ok}. A value of {down} indicates that the last ping to the device was not answered (the "Print CommStatus" command is used as the ping message).
- Enabled – Enables or Disables communication to the associated device from AC256 Driver
- Fault Cause – if the "Status" property value is {fault}, the fault cause is displayed here.
- Heath – contains metadata about the health of this device on the network:

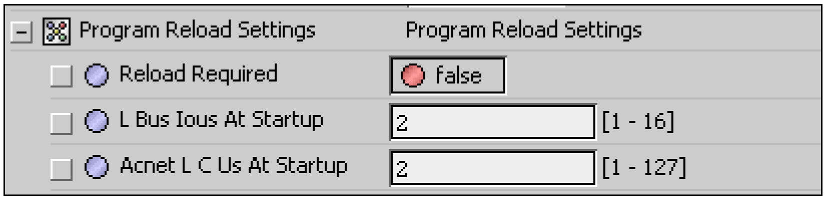
  - Down – indicates if this device is down – should be false under normal operation.
  - Alarm – indicates if this device is in alarm – should be false under normal operation
  - Last OK Time – the last time of successful communication to this device
  - Last Fail Time – the last time of unsuccessful communication to this device
  - Last Fail Cause – the reason of the last communication failure
- Alarm Source Info – configuration items for alarms generated from the ping process (device up and device down events)
- **Device Password –** This is usually formed from the serial number tag inside of the AC256 controller enclosure.
- **Built In User Password –** This should be set to the built in user password that is factory set. Some well known passwords are “Santa”, “bugger”, and “froboz”.
- **Comm Initialized** – indicates a successful login or not
- **Prompt Status –** Indication of the context with which the driver is currently talking to the panel. Normally indicates “R Prompt”
- **Current Domain –** Indication of the current “domain” with which the driver is currently talking to the panel. Normally indicates “0”.
- **Print Status Readings –** Whenever a “status” message is received (usually happens at startup), this property is updated with the values from message. To force an update of the readings, you may issue the “Status Command” action by right clicking on the Andover Device component in the Nav tree.



Print Status Readings	Print Status Readings
□ ○ Unit Number	1
□ ○ Number Attached I O Us	2
□ ○ Iou Comm Faults	1, 2
□ ○ Iou Overrides	
□ ○ Number L C Us	2
□ ○ L C U Comm Faults	1
□ ○ Disabled Var Classes	F, V
□ ○ Number Failures	1
□ ○ Andover Date Time	2007 JUN 18 MON 16 20 6

- **Jace Date Time –** A display of the Jace’s real time clock. This value is also updated as a courtesy (so the user can compare the AC256 time with the Jace time) whenever the “Status Command” action is invoked on the Andover Device.
- **Automatically Set Time –** If set to “true”, then the driver will send a time update command to the AC256 panel on a schedule determined by the “Sync Time Frequency” property.
- **Sync Time Frequency –** how often to send the Jace’s clock time to the Andover panel to keep them time-synchronized.

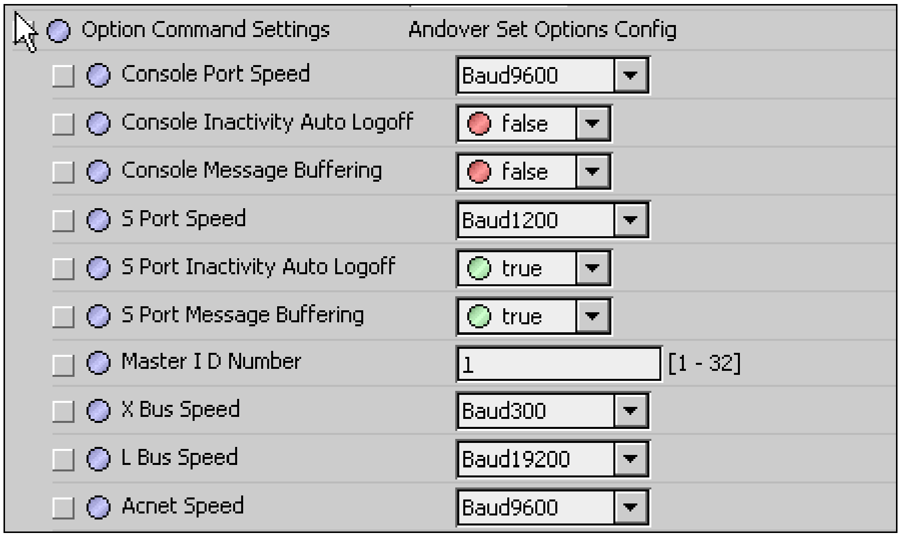
- **Automatically Reload Program –** The driver has the ability to download (automatically, when it detects the panel's memory has been lost) a complete database backup copy. The backup copies are stored on the Jace in a "backups" folder, using panel serial number, domain, and time in the file name.
- **Program Reload Settings –** Settings which need to be entered whenever an AC256 panel is reloaded from scratch. The panel normally will prompt the terminal user for number of IOU's and number of LCU's that need to be configured in to the system.



Program Reload Settings	Program Reload Settings
Reload Required	false
L Bus Ious At Startup	2 [1 - 16]
Acnet L C Us At Startup	2 [1 - 127]

- Reload Required – Driver determines if a reload is required
- L Bus Ious At Startup – Enter the number of IOUs that need to be re-configured into the system, should the controller database need to be reloaded.
- Acnet L C Us At Startup – Enter the number of LCU's that need to be re-configured into the system, should the controller database need to be reloaded.

- **Automatically Set Options** – The driver has the ability to execute the "Set Options" command after a reload if this property is set to "true". The Set Options command prompts the terminal user for several pieces of information. The answers to these prompts should be configured into the property "Option Command Settings" below:
- **Option Command Settings** – The answers to the prompts presented to a terminal user (the driver becomes the terminal user in this case) in response to the "Set Options" command. Note that these values can also be set via the "Andover Backup Manager" view using the "Set Options" button.



Option Command Settings	Andover Set Options Config
Console Port Speed	Baud9600
Console Inactivity Auto Logoff	false
Console Message Buffering	false
S Port Speed	Baud1200
S Port Inactivity Auto Logoff	true
S Port Message Buffering	true
Master I D Number	1 [1 - 32]
X Bus Speed	Baud300
L Bus Speed	Baud19200
Acnet Speed	Baud9600

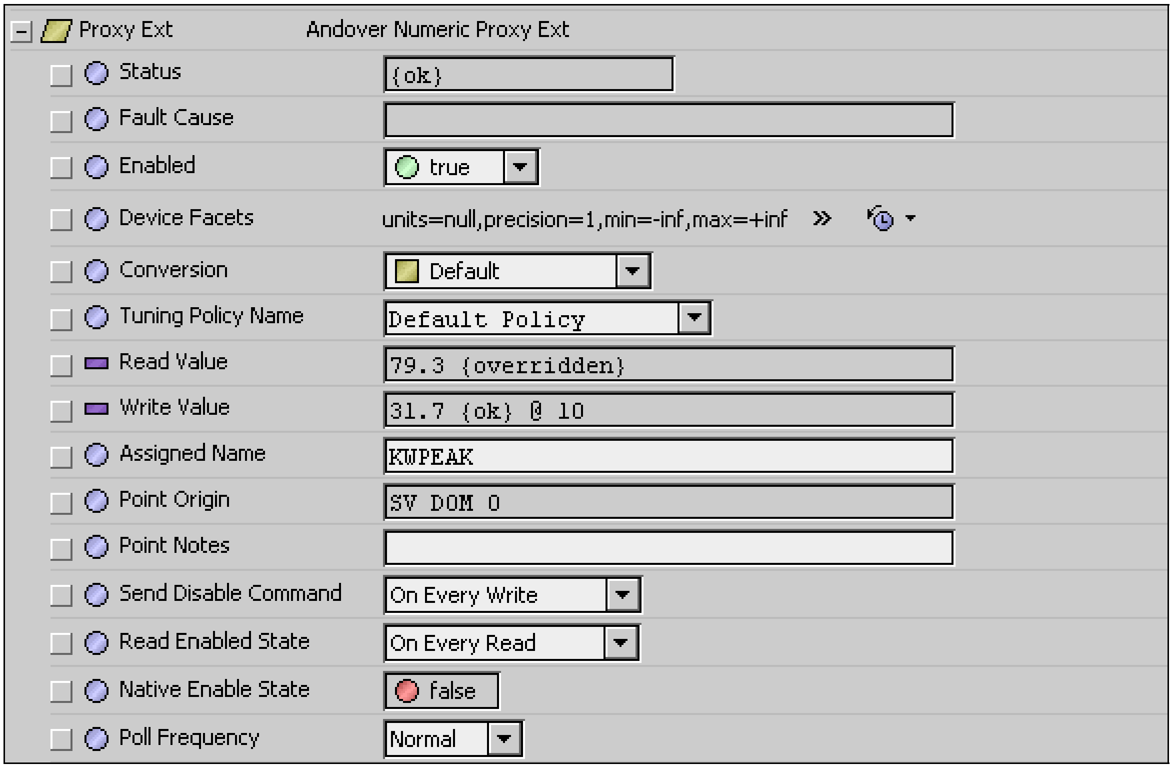
- **Monitor Point –** This is the point that is used in the "ping" process. In order for the panel to be detected as "up", this point must exist in the panel. Defaults to "F1" because this flag point is found in AC256 panels.

- **Points** – A container object that contains proxy points corresponding to points or system variables on the parent AndoverDevice. These proxy points are the data items in this device which need to be polled for data.
- **Use S Port** – Normally, the Jace will be connected to the Andover panel through the “C Port” of the Andover panel. In some installations, it may be required to use the “S Port”. In these cases, set this property to true.(Note: This affects whether or not a newline character is transmitted after the device password is sent during a logon. If not set correctly, logon will be unsuccessful)

AndoverProxyExt

The AC256 proxy extension types take on the readable-writable personality of the control point they are attached to. For example, a AndoverProxyExt, when used as an extension on a NumericPoint has “read only” functionality, but when used on as an extension on a NumericWritable can read and write the point values.

The AndoverProxyExt is the "point-level" component in the NiagaraAX architecture. The following shows an AndoverProxyExt on a NumericWritable control point:



Proxy Ext	Andover Numeric Proxy Ext
Status	{ok}
Fault Cause	
Enabled	true
Device Facets	units=null,precision=1,min=-inf,max=+inf
Conversion	Default
Tuning Policy Name	Default Policy
Read Value	79.3 {overridden}
Write Value	31.7 {ok} @ 10
Assigned Name	KWPEAK
Point Origin	SV DOM 0
Point Notes	
Send Disable Command	On Every Write
Read Enabled State	On Every Read
Native Enable State	false
Poll Frequency	Normal

NOTE: In the following properties, the grayed out properties are inherited from the base NiagaraAX driver classes, and as such are only touched on here. For a full explanation, refer to the “Driver architecture” chapter in the NiagaraAX-3.x Drivers Guide.

- Status – The status of the proxy ext. Will normally be {ok}. Will transition to ‘fault’ if a response to a poll is detected to be an error. Will transition to ‘down’ if the parent device is down. Will indicate ‘stale’ if the point value has not been updated within the stale timeout period (see Tuning Policy property on the network).
- Fault Cause – if the “Status” property value is {fault}, the fault cause is displayed here.

- Enabled – Enables or Disables communication to the associated point from Andover Driver
- Device Facets – Used to modify the value read to the value displayed. Typically not used in an AC256 integration.
- Conversion – Can apply custom conversion to the point values. Applies to both read and (inversely) to write values.
- Tuning Policy Name – selects the tuning policy to apply to this point. See "Tuning Policies" property of the AndoverNetwork
- Read Value – the value last read from the point in the panel
- Write Value – the value last written from this proxy ext (does not apply to read only point types)
- **Assigned Name** – The name used to address this point in the Andover panel. For AC256, this is the only parameter necessary to specify a unique point.
- **Point Origin –** If the point was added using the point manager, then this property will contain the actual location of the point in the AC256 network. For this example, "SV DOM 0" indicates the point came from the SV table of Domain 0 (the AC256 panel). This property is for informational use only, and does not affect the operation of the driver.
- **Point Notes –** If the point was added using the point manager, this property *ma*y contain additional meta data. For example, if the point is located on an LCU or IOU, the wiring terminals will be displayed by default. The integrator is free to add, modify, or delete any text in this property – it is for informational use only and does not affect the operation of the driver.
- **Send Disable Command –** For points that are Writable, in order for the driver's written value to "stick" in the panel, the point must be in a "Disabled" state in the panel. This allows the terminal user (the driver) to set a value without having the Andover controller's program over-write the value from its internal control logic. There are three selections here:
    - **Never –** This setting assumes that either the point is already disabled in the panel, or, alternatively, that the point may or may not be enabled, but is never updated internally by the Andover panel. This setting is the most efficient use of comm. bandwidth because a separate "Disable" command is not required.
    - **On Every Write –** The "Disable" command will be sent EVERY time a write is required to the point. This is the safest setting to assure that the value written will be the value retained in the Andover panel, but has the tradeoff of consuming the most additional comm. bandwidth because two messages are required to be sent for one value written (the 'disable' command, followed by the 'modify' command)
    - **Only On First Write –** This setting represents a compromise between never sending disable, and always sending disable. The 'disable' command will only be sent on the first write, and the driver will "remember" that it has already disabled the point on additional writes. The possibility exists that the driver and panel can become "out of sync" if there is another entity (external to the driver) that also sets the disabled state of the point (NOTE: it is possible to do so from program logic in the panel).

- **Read Enabled State –** This is the “read” equivalent of the “send disable command” property detailed above. This allows the terminal user (the driver when it is polling) read the enabled/disabled state of a point in the panel and mark it accordingly in the proxy ext. There are three selections:
  - **Never –** For points for which you don’t care what the enabled/disabled state is. This has the advantage of consuming the least bandwidth, but the disadvantage of not knowing what the point state is.
  - **On Every Read –** For points for which it is critical to know the disabled/enabled state. This setting consumes more bandwidth (must perform 2 reads – one for state, one for value) whenever polling the point.
  - **Only On First Read –** Compromise setting. The driver will read the enabled state the first time it polls, and “remember” that state from then on out. For writable points, a write which includes a “disable” or a return to “auto” from a previously disabled point will also update the remembered value.
- **Native Enable State –** The enabled/disabled state of the point in the Andover panel. This is the “remembered” state which is updated either from a read (if reading of state is also enabled) or from a write (if writing of state is enabled).
- **Poll Frequency** – Fast, Normal, or Slow. Rates are determined by the Poll Scheduler settings on the AndoverNetwork/Communicator/PollScheduler property.

Andover AC256 Views

Andover AC256 views include the following:

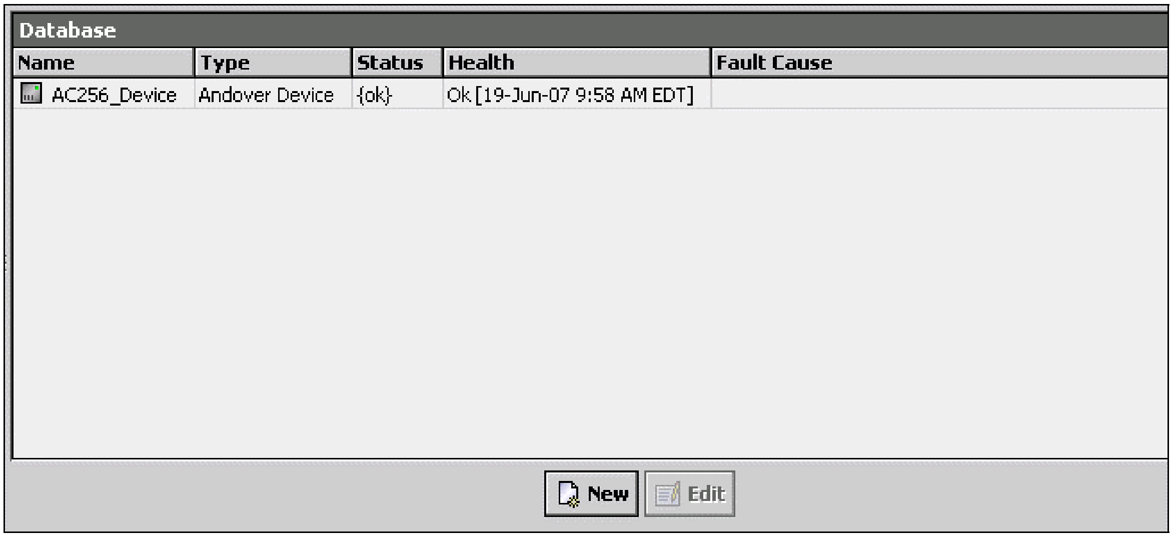
- Device Manager
- Point Manager
- AC256 Virtual Terminal

Device Manager

The Device Manager is the default view when you double-click on an AndoverNetwork in the Nav tree. This manager view provides a quick and easy way to display and add AC256 devices.

WARNING: ONLY ADD ONE and ONLY ONE device object!!!

The Andover AC256 Device Manager consists of either one pane. There is no “discover” functionality for AC256, since one and only one device is allowed per network. The view below shows a typical Andover AC256 Device Manager view.



Database

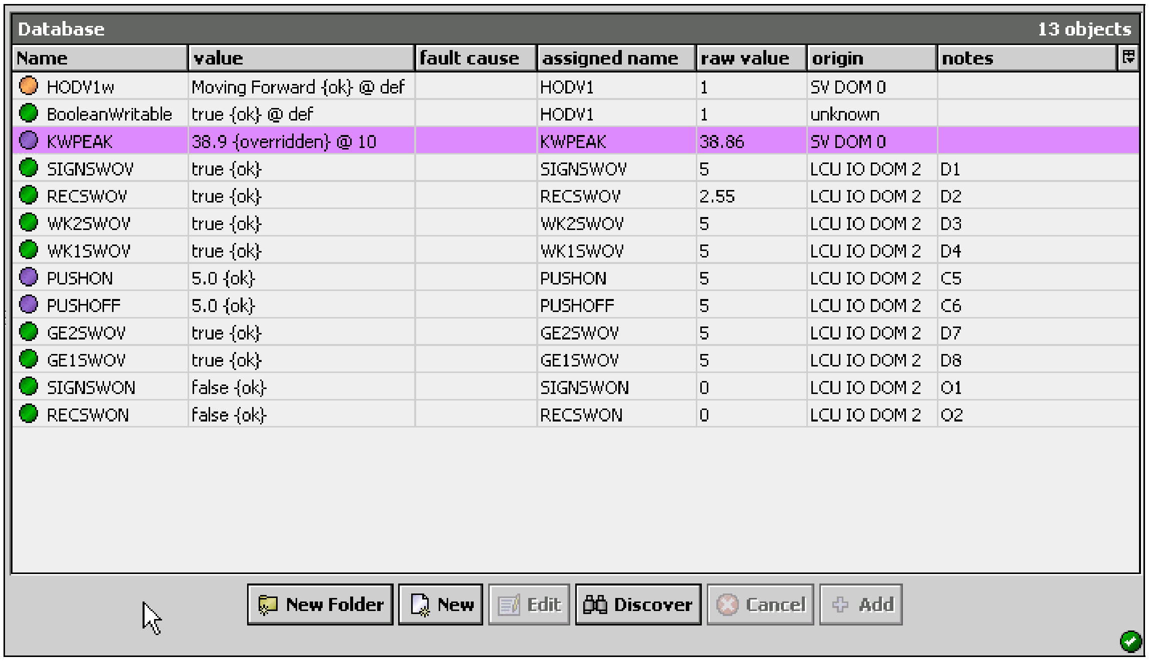
Name	Type	Status	Health	Fault Cause
AC256_Device	Andover Device	{ok}	Ok [19-Jun-07 9:58 AM EDT]	

Click the “New” button to add the Andover Device to the network container. Remember, ONLY ADD ONE DEVICE.

Click the “Edit” button to bring up a dialog box whereby you can change the name or the enabled state of the device.

Point Manager

The Point Manager is the default view when you double-click on a “points” folder under an AndoverDevice in the Nav tree. The Point Manager is a table-based view, where each row represents a unique point in an AC256 controller. Below is an example Point Manager view:

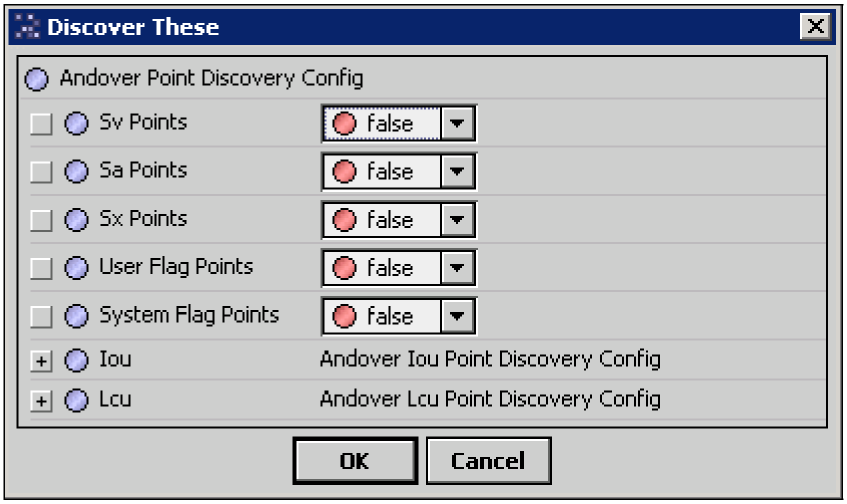


Database 13 objects

Name	value	fault cause	assigned name	raw value	origin	notes
HODV1w	Moving Forward {ok} @ def		HODV1	1	SV DOM 0	
BooleanWritable	true {ok} @ def		HODV1	1	unknown	
KWPEAK	38.9 {overridden} @ 10		KWPEAK	38.86	SV DOM 0	
SIGNSWOV	true {ok}		SIGNSWOV	5	LCU IO DOM 2	D1
RECSWOV	true {ok}		RECSWOV	2.55	LCU IO DOM 2	D2
WK2SWOV	true {ok}		WK2SWOV	5	LCU IO DOM 2	D3
WK1SWOV	true {ok}		WK1SWOV	5	LCU IO DOM 2	D4
PUSHON	5.0 {ok}		PUSHON	5	LCU IO DOM 2	C5
PUSHOFF	5.0 {ok}		PUSHOFF	5	LCU IO DOM 2	C6
GE2SWOV	true {ok}		GE2SWOV	5	LCU IO DOM 2	D7
GE1SWOV	true {ok}		GE1SWOV	5	LCU IO DOM 2	D8
SIGNSWON	false {ok}		SIGNSWON	0	LCU IO DOM 2	O1
RECSWON	false {ok}		RECSWON	0	LCU IO DOM 2	O2

The “**New Folder**”, “**New**”, and “**Edit**” buttons are not unique to the AC256 point Manager, and are explained in the “Driver architecture” chapter in the *NiagaraAX-3.x Drivers Guide*.

By clicking the “**Discover**” button, the “learn” mode of the manager is invoked (the panes will be split, and a “discovery” table will be displayed in the top pane) and a dialog box will be popped up to prompt the user for some additional data to guide the discovery process, as shown here:



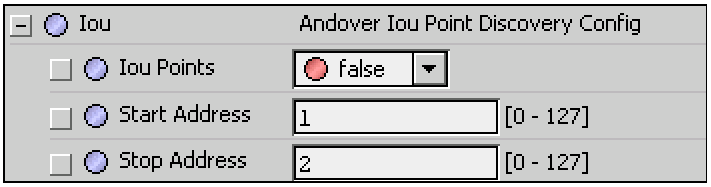
Discover These

Andover Point Discovery Config

□ Sv Points: false
□ Sa Points: false
□ Sx Points: false
□ User Flag Points: false
□ System Flag Points: false
+ Iou: Andover Iou Point Discovery Config
+ Lcu: Andover Lcu Point Discovery Config

This dialog box contains several parameters that determine which points in the system are to be discovered:

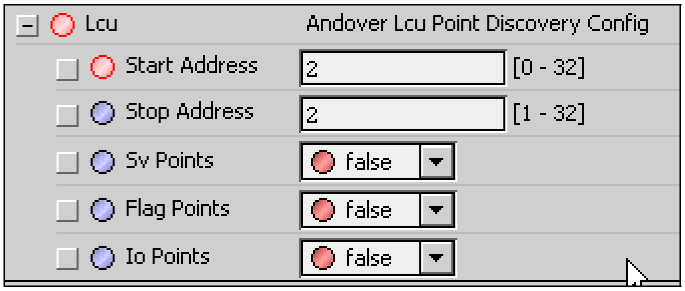
- **Sv Points –** Discover points from the AC256 controller's SV table that have been assigned names. These points are obtained by sending the "Print SV" command to the controller (domain 0)
- **Sa Points –** Discover points from the AC256 controller's SA table that have been assigned names. These points are obtained by sending the "Print SA" command to the controller (domain 0)
- **Sx Points –** Discover points from the AC256 controller's SX table that have been assigned names. These points are obtained by sending the "Print SX" command to the controller (domain 0)
- **User Flag Points –** Discover points from the AC256 controller's flags table that have been defined programmatically (i.e., not predefined flags). These points are obtained by sending the command "Print Flags" to the controller at domain 0, and looking for flag points that occur before the "`System Defined Flags`" section.
- **System Flag Points –** Discover points from the AC256 controller's flags table that are pre-defined. These points are obtained by sending the command "Print Flags" to the controller at domain 0, and looking for flag points that occur after the "`System Defined Flags`" section.
- **IOU –** Expand this section to refine which points are to be learned from any IOU's that may be attached to the AC256 controller:



Iou	Andover Iou Point Discovery Config
Iou Points	false
Start Address	1 [0 - 127]
Stop Address	2 [0 - 127]

  - **Iou Points** – Set to true to learn IOU points. IOUs numbered starting at "Start Address" through "Stop Address" will be learned. Points with assigned names from these panels will be learned by sending the command "Print *n*", where *n* is the IOU address.
  - **Start Address** – Enter the starting IOU panel address to learn
  - **Stop Address –** Enter the last IOU panel address to learn

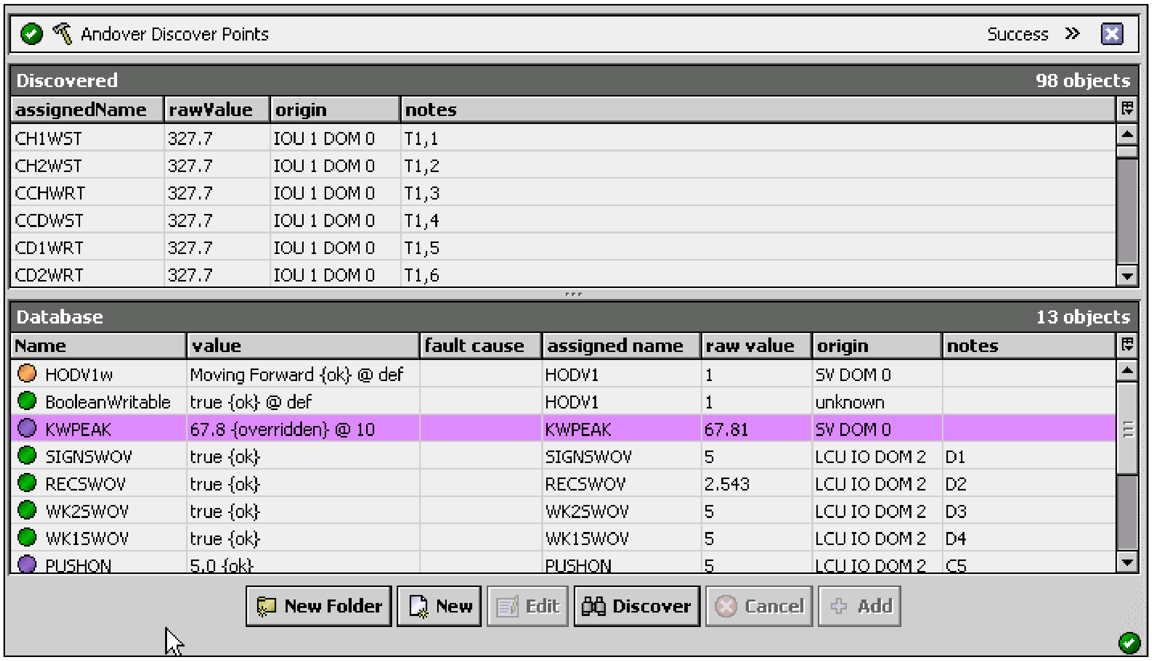
- **LCU –** Expand this section to refine which points are to be learned from any LCU's that may be attached to the AC256 controller:



Lcu	Andover Lcu Point Discovery Config
Start Address	2 [0 - 32]
Stop Address	2 [1 - 32]
Sv Points	false
Flag Points	false
Io Points	false

- **Start Address –** Enter the address of the starting LCU to learn from. This is equivalent to the “domain” number of the first LCU.
- **Stop Address –** Enter the address of the last LCU to learn from. This is equivalent to the “domain” number of the last LCU.
- **Sv Points –** Set to true to learn assigned points from the SV table of the LCU. Points are learned by sending the “Dom *n*” command and then the “Print SV” command, where *n* is the LCU address.
- **Flag Points –** Set to true to learn assigned points from the flags table of the LCU. Points are learned by sending the “Dom *n*” command and then the “Print Flags” command, where *n* is the LCU address
- **Io Points –** Set to true to learn assigned points from the IO table of the LCU. Points are learned by sending the “Dom *n*” command and then the “Print 1” command, where *n* is the LCU address

Clicking the OK will cause a learn process to be kicked off, the progress of which will be displayed in the job bar at the top of the split screen. When the learn process is done, the results will be displayed in the “Discovered” pane on the top half of the point manager display:



Discovered — 98 objects

assignedName	rawValue	origin	notes
CH1WST	327.7	IOU 1 DOM 0	T1,1
CH2WST	327.7	IOU 1 DOM 0	T1,2
CCHWRT	327.7	IOU 1 DOM 0	T1,3
CCDWST	327.7	IOU 1 DOM 0	T1,4
CD1WRT	327.7	IOU 1 DOM 0	T1,5
CD2WRT	327.7	IOU 1 DOM 0	T1,6

Database — 13 objects

Name	value	fault cause	assigned name	raw value	origin	notes
● HODV1w	Moving Forward {ok} @ def		HODV1	1	SV DOM 0	
● BooleanWritable	true {ok} @ def		HODV1	1	unknown	
● KWPEAK	67.8 {overridden} @ 10		KWPEAK	67.81	SV DOM 0	
● SIGNSWOV	true {ok}		SIGNSWOV	5	LCU IO DOM 2	D1
● RECSWOV	true {ok}		RECSWOV	2.543	LCU IO DOM 2	D2
● WK2SWOV	true {ok}		WK2SWOV	5	LCU IO DOM 2	D3
● WK1SWOV	true {ok}		WK1SWOV	5	LCU IO DOM 2	D4
● PUSHON	5.0 {ok}		PUSHON	5	LCU IO DOM 2	C5

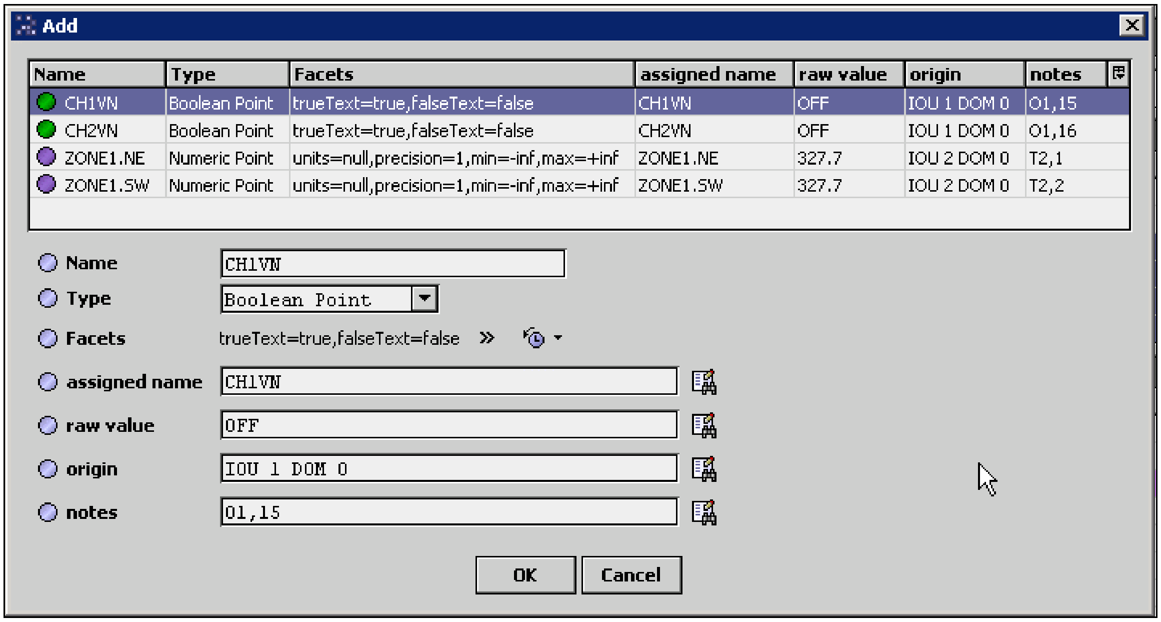
The display line for a learned point shows the point name and several columns that are the information returned from the various commands sent to the controller during the learn.

- assignedName: The name of the point as it exists in the Andover system.
- rawValue: The value of the point as reported by the Andover system.
- origin: During the learn process, the driver keeps track of what component or table it is requesting point data from. This information is recorded as the “origin” of the point.

- notes: If additional information was returned, this information is recorded here for informational purposes only. The notes value is transferred to any point that is added as a result of the learn, but may be edited/deleted after the point has been added to the database. This column value may or may not be blank.

To add one or more points to the database, select them (multi-select by using CTRL-click or SHIFT-click operations with the mouse), and then select Add.

By clicking "Add" the point(s) are displayed in an "Add" dialog box where the added name, point name, and point type may be modified:



Name	Type	Facets	assigned name	raw value	origin	notes
CH1VN	Boolean Point	trueText=true,falseText=false	CH1VN	OFF	IOU 1 DOM 0	01,15
CH2VN	Boolean Point	trueText=true,falseText=false	CH2VN	OFF	IOU 1 DOM 0	01,16
ZONE1.NE	Numeric Point	units=null,precision=1,min=-inf,max=+inf	ZONE1.NE	327.7	IOU 2 DOM 0	T2,1
ZONE1.SW	Numeric Point	units=null,precision=1,min=-inf,max=+inf	ZONE1.SW	327.7	IOU 2 DOM 0	T2,2

○ Name: CH1VN
○ Type: Boolean Point
○ Facets: trueText=true,falseText=false
○ assigned name: CH1VN
○ raw value: OFF
○ origin: IOU 1 DOM 0
○ notes: 01,15

In this dialog, the points may be edited individually or in a batch. Note that any edits apply to ALL selected (highlighted) points in the Add dialog box. The "Type" determines which base point type will be created. ***There may be reasons for changing the Type at this point, but in general the most appropriate type of base point is already selected as the first type in the "Type" pull down selection box. You might want to verify that the Type does not need to be changed to a "writable" version of the same type.*** For example, you may want an "enumeration" point type instead of a "numeric" point type for a point which returns an integer number.

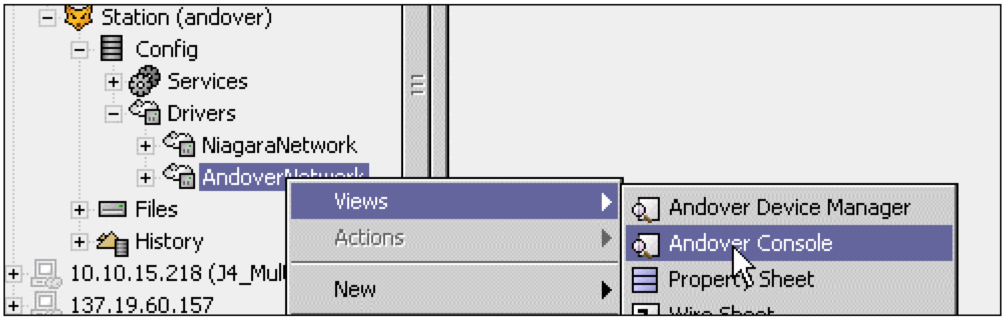
The "name" of the point to be added is derived from the "assignedName" of the point as learned from the controllers. Niagara will not allow duplicate names, so if you attempt to learn and add the same point multiple times, Niagara will modify the name so that it is unique. You may elect to edit the point "name" to any valid Niagara name.

Once the point(s) are satisfactorily edited, click "**OK**" to create the proxy points corresponding to the group addresses.

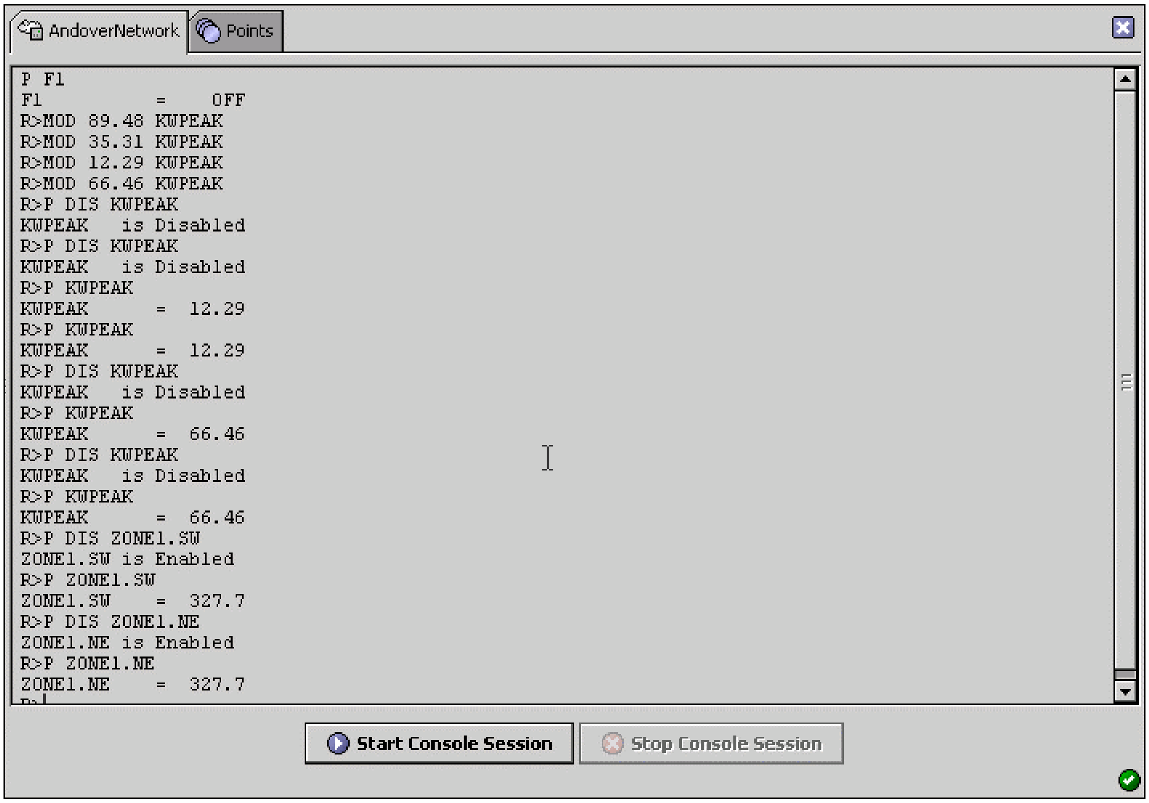
Once added to the database, the points appear in the lower pane (or only pane if not in learn mode) of the Point Manager display.

AC256 Virtual Terminal

The AC256 Console is a special view under the AndoverNetwork component. It is reached by right-clicking the Andover Network component in the component Nav tree, selecting "Views" and then "Andover Console":



The console view allows the user to monitor the driver activity as it is happening in real time. In addition, the user can interrupt the normal driver communications to perform interactive terminal commands with the AC256 system:



```
P F1
F1          =    OFF
R>MOD 89.48 KWPEAK
R>MOD 35.31 KWPEAK
R>MOD 12.29 KWPEAK
R>MOD 66.46 KWPEAK
R>P DIS KWPEAK
KWPEAK   is Disabled
R>P DIS KWPEAK
KWPEAK   is Disabled
R>P KWPEAK
KWPEAK      =  12.29
R>P KWPEAK
KWPEAK      =  12.29
R>P DIS KWPEAK
KWPEAK   is Disabled
R>P KWPEAK
KWPEAK      =  66.46
R>P DIS KWPEAK
KWPEAK   is Disabled
R>P KWPEAK
KWPEAK      =  66.46
R>P DIS ZONE1.SW
ZONE1.SW is Enabled
R>P ZONE1.SW
ZONE1.SW    =  327.7
R>P DIS ZONE1.NE
ZONE1.NE is Enabled
R>P ZONE1.NE
ZONE1.NE    =  327.7
```

The AC256 console view shows the terminal interface of the AC256 device to which driver is connected. It shows the real-time communications that are taking place between the driver and the controller to the panel.

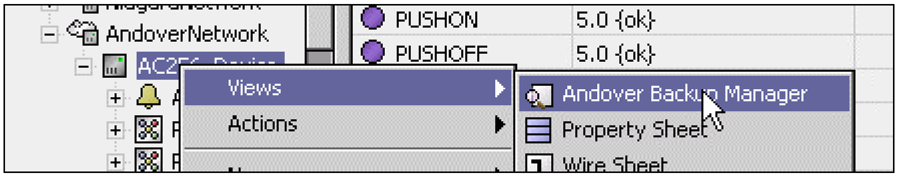
In addition, a user can enter an interactive session with the controller by clicking the "Start Console Session" button. The session will remain active until the user clicks "Stop Console Session" or exits the view. A console session will pre-empt all other communications for the duration of the session, causing the devices to be marked down and point to go stale. Exiting the console session should allow the driver to resume normal driver communications.

While in the console session, all keystrokes are from the user are sent to the AC256 controller just as if a terminal were connected directly to the controller. Therefore, the full range of AC256 configuration, programming, and file operations are available while in interactive mode.

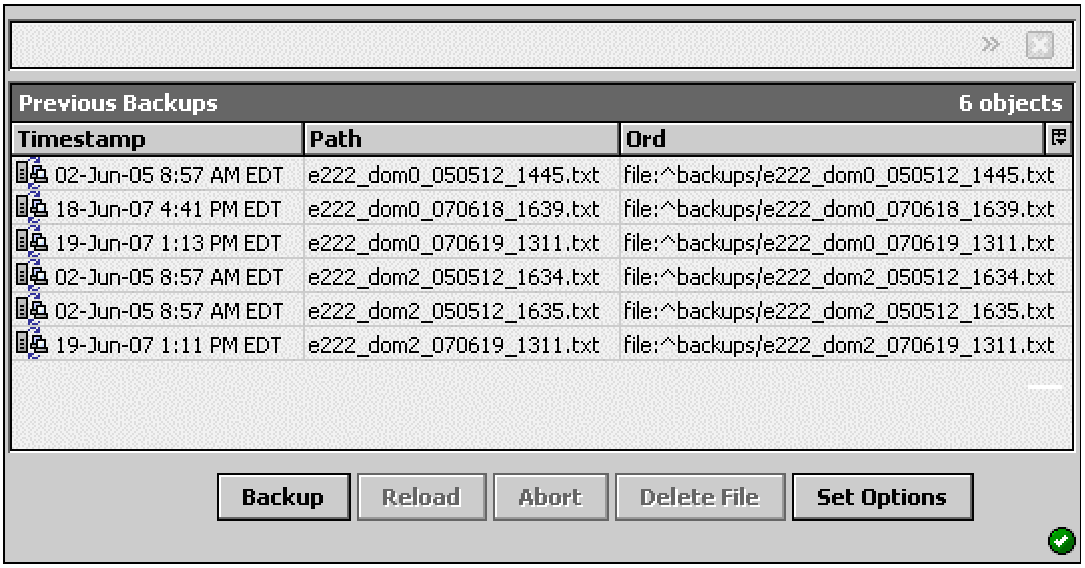
> ***CAUTION**: Care should be used to leave the controller in command line mode at an "R>" prompt when exiting the console session. Failure to do so may prevent the driver from resuming normal driver activity.*

Backup Manager View

A special view is provided on the Andover Device object which allows a user to perform a backup or restore of the AC256 main controllers or LCUs. The backup manager is accessed by right clicking on the Andover Device in the nav tree, selecting "Views", and then selecting "Andover Backup Manager":



A typical backup manager view is shown here:

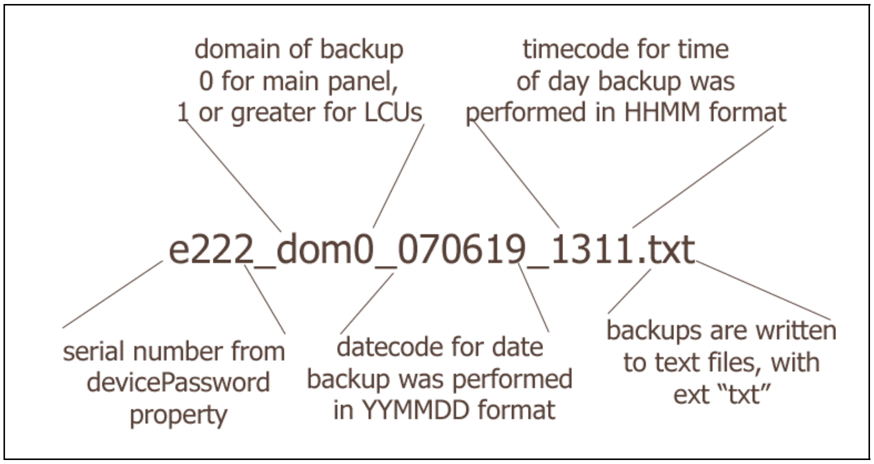

Previous Backups 6 objects

Timestamp	Path	Ord
02-Jun-05 8:57 AM EDT	e222_dom0_050512_1445.txt	file:^backups/e222_dom0_050512_1445.txt
18-Jun-07 4:41 PM EDT	e222_dom0_070618_1639.txt	file:^backups/e222_dom0_070618_1639.txt
19-Jun-07 1:13 PM EDT	e222_dom0_070619_1311.txt	file:^backups/e222_dom0_070619_1311.txt
02-Jun-05 8:57 AM EDT	e222_dom2_050512_1634.txt	file:^backups/e222_dom2_050512_1634.txt
02-Jun-05 8:57 AM EDT	e222_dom2_050512_1635.txt	file:^backups/e222_dom2_050512_1635.txt
19-Jun-07 1:11 PM EDT	e222_dom2_070619_1311.txt	file:^backups/e222_dom2_070619_1311.txt

> *NOTE:* The manager presents a list of backups that currently exist **on the Jace**. Backups are always stored on the Jace because the driver has the ability to automatically reload the controller database in the event that the controller ever loses its database. However, the tradeoff is that the user must be aware that each backup consumes valuable storage space on the Jace.

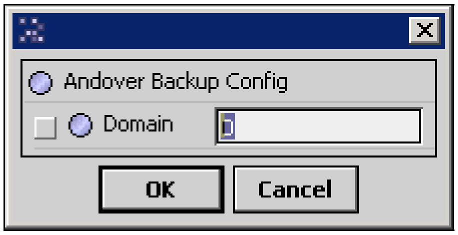
If desired, a user with sufficient privileges can use the Niagara Workbench platform tools to copy these backups to a Workbench PC for archival purposes.

Files are automatically named by the backup manager using a set of rules. See the following example:



Performing a Backup

A “backup” of the AC256 system can be performed by clicking on the “Backup” button. This causes a dialog box to appear:



The user should enter the “domain” of the Andover device they wish to backup. An entry of “0” backups up the main AC256 panel, and an entry of 1 or greater corresponds to addresses of LCUs which are attached to the AC256.

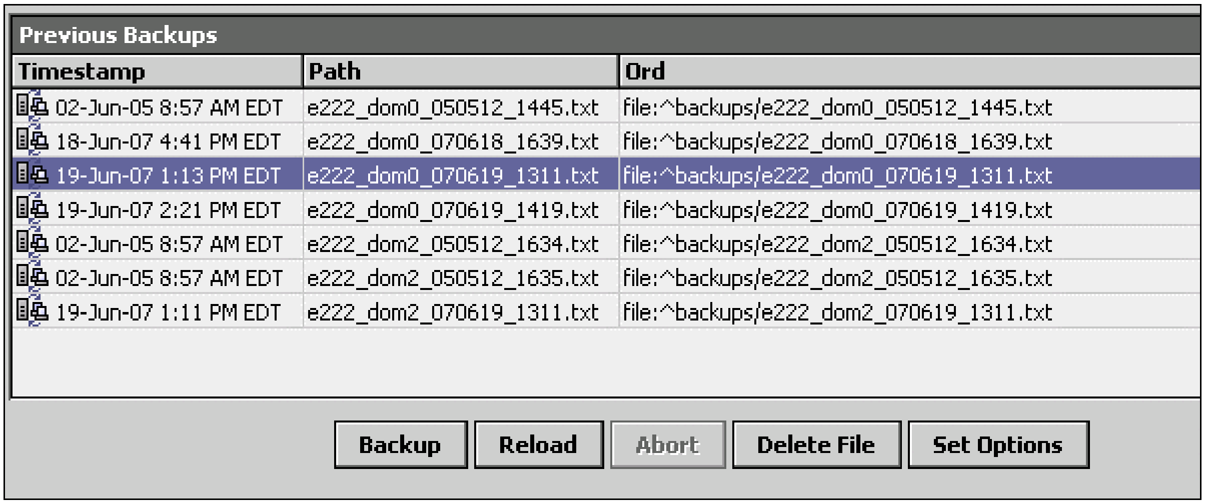
Once the user clicks OK, the backup will be started, and the “Abort” button will be the only button enabled. To abort the backup, click “Abort”.

The driver waits a considerable amount of time to be certain that all other comm. Activity has ceased. It then switches to the domain of choice and sends the “dump” command to the AC256 network.

PLEASE BE PATIENT….When the backup operation finishes, the buttons should return to the default state.

Performing a Reload

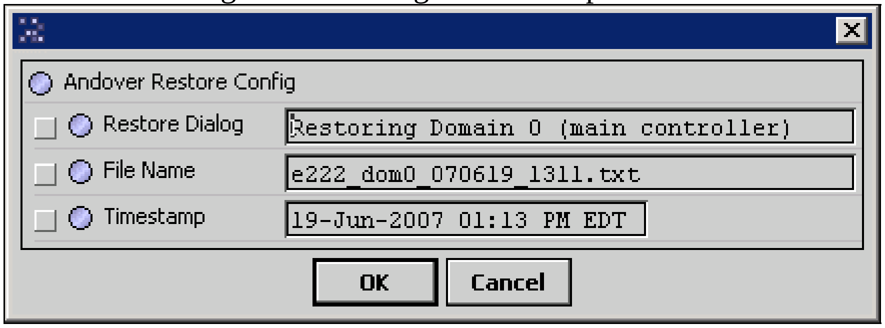
A “reload” of a controller is invokes by highlighting a single file in the file list (which then enables the “Reload” button) , and then selecting the “Reload” button:



Previous Backups

Timestamp	Path	Ord
02-Jun-05 8:57 AM EDT	e222_dom0_050512_1445.txt	file:^backups/e222_dom0_050512_1445.txt
18-Jun-07 4:41 PM EDT	e222_dom0_070618_1639.txt	file:^backups/e222_dom0_070618_1639.txt
19-Jun-07 1:13 PM EDT	e222_dom0_070619_1311.txt	file:^backups/e222_dom0_070619_1311.txt
19-Jun-07 2:21 PM EDT	e222_dom0_070619_1419.txt	file:^backups/e222_dom0_070619_1419.txt
02-Jun-05 8:57 AM EDT	e222_dom2_050512_1634.txt	file:^backups/e222_dom2_050512_1634.txt
02-Jun-05 8:57 AM EDT	e222_dom2_050512_1635.txt	file:^backups/e222_dom2_050512_1635.txt
19-Jun-07 1:11 PM EDT	e222_dom2_070619_1311.txt	file:^backups/e222_dom2_070619_1311.txt

A “Reload Configuration” dialog box then is presented.

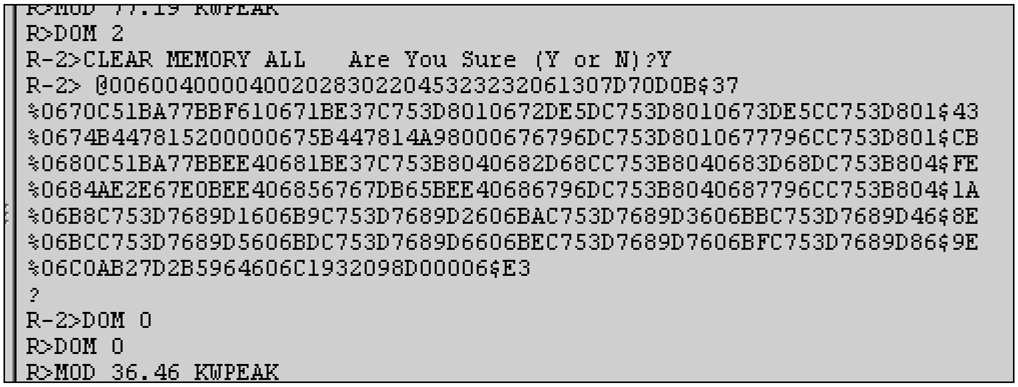


Andover Restore Config

□ Restore Dialog	Restoring Domain 0 (main controller)
□ File Name	e222_dom0_070619_1311.txt
□ Timestamp	19-Jun-2007 01:13 PM EDT

If the user clicks on the “OK” button, a restore of a previous backup file will be performed.

NOTE: A restore takes considerably longer to perform than a backup. If you wish, you may switch to the “Andover Console” view to monitor the progress of the backup and to verify that everything is still progressing. See following example:



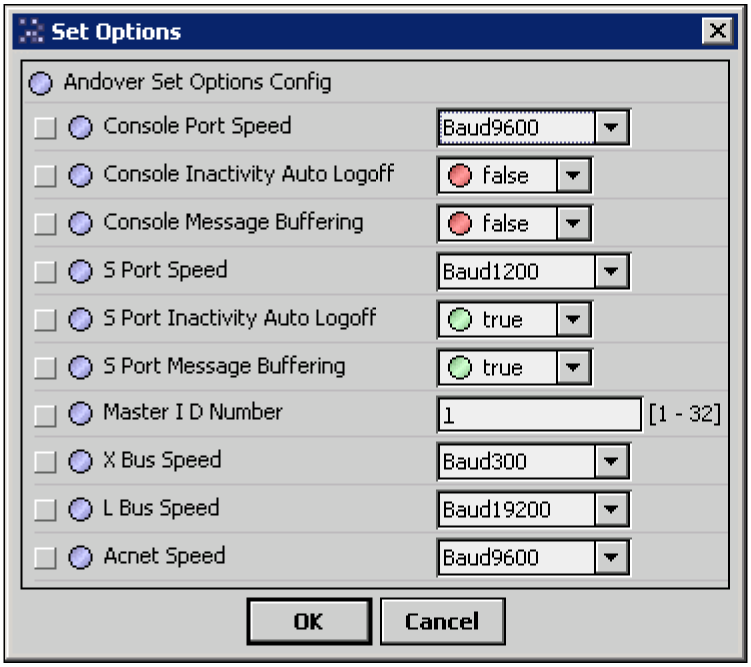
```
R>MOD 77.19 KWPEAK
R>DOM 2
R-2>CLEAR MEMORY ALL   Are You Sure (Y or N)?Y
R-2> @006004000040020283022045323232061307D70D0B$37
%0670C51BA77BBF610671BE37C753D8010672DE5DC753D8010673DE5CC753D801$43
%0674B447815200000675B447814A98000676796DC753D8010677796CC753D801$CB
%0680C51BA77BBEE40681BE37C753B8040682D68CC753B8040683D68DC753B804$FE
%0684AE2E67E0BEE406856767DB65BEE40686796DC753B8040687796CC753B804$1A
%06B8C753D7689D1606B9C753D7689D2606BAC753D7689D3606BBC753D7689D46$8E
%06BCC753D7689D5606BDC753D7689D6606BEC753D7689D7606BFC753D7689D86$9E
%06C0AB27D2B5964606C1932098D00006$E3
?
R-2>DOM 0
R>DOM 0
R>MOD 36.46 KWPEAK
```

Setting Options on the Controller

The Backup Manager view has one additional button called “Set Options”. This is a convenience to allow a user to invoke the “Set Options” command on the AC256 controller. Normally (i.e., from a terminal session) the Set Options command will prompt the terminal user for several parameters to enter. Invoking this command from the Backup Manager automatically answers all of the prompts with data stored in the AndoverDevice’s “Option Command Settings” property.

NOTE: This command is also automatically invoked if configured and if needed after an automatic reload of the controller database.

Clicking the Set Options button causes the following dialog box to be displayed:



Andover Set Options Config

□ Console Port Speed	Baud9600
□ Console Inactivity Auto Logoff	false
□ Console Message Buffering	false
□ S Port Speed	Baud1200
□ S Port Inactivity Auto Logoff	true
□ S Port Message Buffering	true
□ Master I D Number	1 [1 - 32]
□ X Bus Speed	Baud300
□ L Bus Speed	Baud19200
□ Acnet Speed	Baud9600

As stated, the values displayed here are from the “Option Command Settings” property of the Andover Device. Click “OK” to set these options in the AC256 controller

Point Value Conversion Considerations

Conversion from AC256 point values to Niagara Boolean points

The AC256 driver uses the following sequence of conversions to convert AC256 point data into Boolean data for use within a BooleanPoint or BooleanWritable. Using this conversion technique it is possible to convert AC256 Boolean, Numeric, or String type points to Boolean values.

1) If raw point value is “0” or “Off”, the value is *false*. If the raw point value is “1” or “On” the value is *true*.

2) Else, try to convert the raw point string to a number (see “Conversion from AC256 point values to Niagara Numeric points” below). If this succeeds, then if the value is non-zero the Niagara value is *true*, else the value is *false*.

3) Else, the value cannot be converted, and will display with a “fault” status.

Conversion from AC256 point values to Niagara Numeric points

The AC256 driver uses the following sequence of conversions to convert AC256 point data into numeric data for use within a NumericPoint or NumericWritable. Using this conversion technique it is possible to convert AC256 Boolean, Numeric, or limited String type points to Numeric values.

1) Try to convert the point string to a number.

2) Else, see if the point string is “-On”, “On” or “Off”. If so, return -1.0, 1.0 or 0.0, respectively.

3) Else, the value cannot be converted, and will display with a “fault” status.

Conversion from AC256 point values to Niagara Enum points

The AC256 driver uses the following sequence of conversions to convert AC256 point data into enum data for use within a EnumPoint or EnumWritable. Using this conversion technique it is possible to convert AC256 Boolean, Numeric, or limited String type points to Enum values.

1) Try to convert the point string to a number. If this succeeds, then use the resulting number (truncated to nearest whole number) as the ordinal into the enumeration

2) Else, try to use the point string directly as the “tag” of the enumeration. If it matches one of the defined tags, then the enumeration value is the value that corresponds to that tag.

3) Else, the value cannot be converted, and will display with a “fault” status.

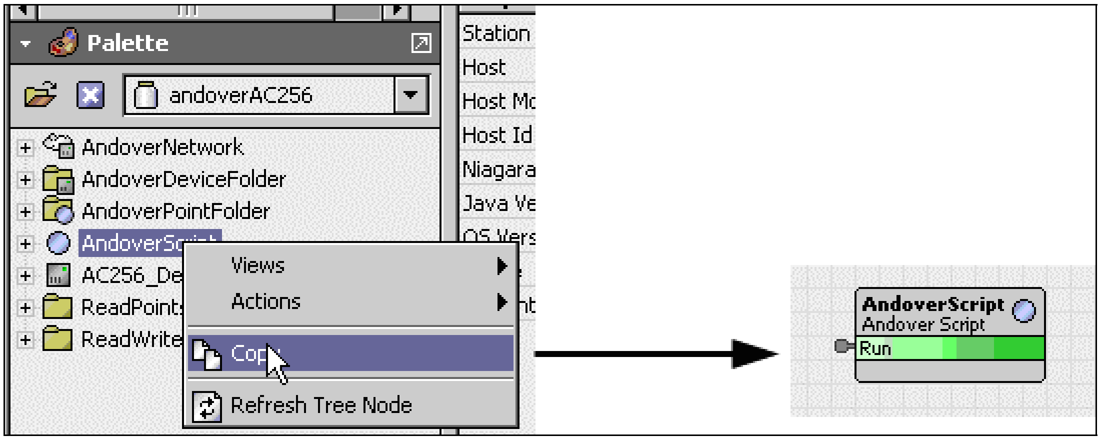
Conversion from AC256 point values to Niagara String points

No conversion is performed. The resultant value of the Niagara point is the string to the right of the “=” sign of the “Raw Response”.

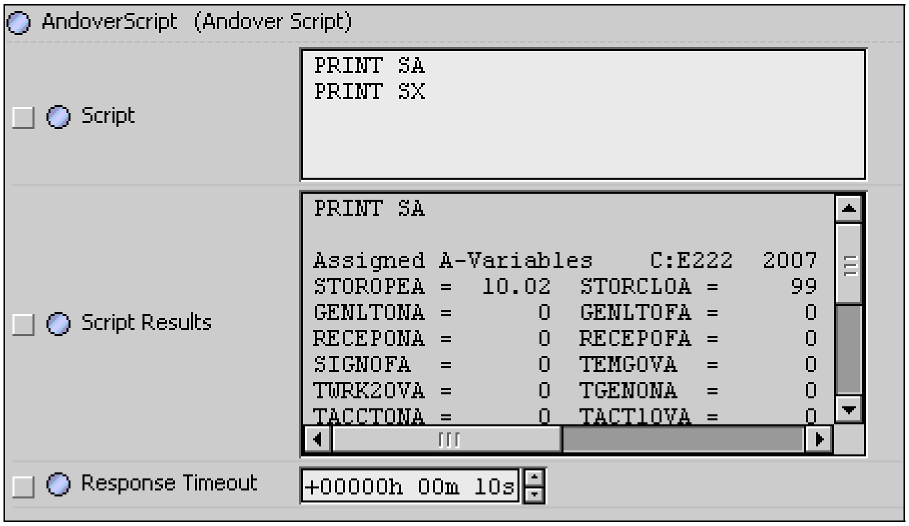
Using the AndoverScript Object

There is a special object called an “AndoverScript” which can be placed under an AndoverDevice object to perform a sequence of un-interrupted commands to the Andover controller. This object can be invoked whenever the “run” action is invoked on the object.

The AndoverScript must be copied and pasted from the AndoverAC256 palette into the wire sheet of the AndoverDevice object:



There are three properties on the AndoverScript object:



AndoverScript (Andover Script)

Script

```
PRINT SA
PRINT SX
```

Script Results

```
PRINT SA

Assigned A-Variables    C:E222   2007
STOROPEA =   10.02  STORCLOA =     99
GENLTONA =       0  GENLTOFA =      0
RECEPONA =       0  RECEPOFA =      0
SIGNOFA  =       0  TEMGOVA  =      0
TWRK20VA =       0  TGENONA  =      0
TACCTONA =       0  TACT1OVA =      0
```

Response Timeout +00000h 00m 10s

- **Script**- This is a list of commands to be sent to the Andover controller whenever this object’s “run” action is invoked.
- **ScriptResults**- This is any text which is returned from the controller as a result of executing the script
- **ResponseTimeout**- This is how long the script will use *after every step* is executed to wait for a response.

In order to invoke the script, the “run” action on the object’s input must be linked to another object. For example, to invoke this script whenever the driver is successful logging on the Andover network, link the Andover Device’s “logonSuccessful” topic to the script’s “run” action. There are many “kit” objects which can also be linked to the “run” action.

© 2008 Tridium, Inc

Document Change Log

- March 19, 2008: Updated to reference the *NiagaraAX Drivers Guide* instead of the *NiagaraAX User Guide*.
- June 26, 2007: Initial draft document.