



NiagaraAX PHP Driver User Guide May 1 2006

This documents usage of the American Auto-Matrix Public Host Protocol (PHP) driver for the NiagaraAX framework.

- COMPATIBILITY2**
 - PHP SPECIFICATIONS2
 - PLATFORMS3
- INSTALLATION4**
- QUICK START5**
- PHP COMPONENT GUIDES6**
 - AAPHPSTARNETWORK6
 - AAPHPSTARDEVICE8
 - AAPHPSTARPROXYEXT10
- PHP VIEWS12**
 - AA PHP STAR DEVICE MANAGER12
 - “Send File”, “Receive File”, and “Examine Setup”*14
 - PHP STAR POINT MANAGER16
 - PHP VIRTUAL TERMINAL VIEW20
- PHP “CV” ATTRIBUTES ON NUMERIC POINTS22**
- EXAMINE SETUP AND EXAMINE DISCOVERY FEATURES24**
 - EXAMINE SETUP24
 - Examine Setup Script*24
 - Examine Setup Usage*26
 - EXAMINE DISCOVERY28

Information and specifications published here are current as of the date of publication of this document. Tridium, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia. Products or features contained herein may be covered by one or more U.S. or foreign patents.

Compatibility

PHP Specifications

This driver is compliant with PHP version 8.10 protocol specifications, as published in the document “PHP version 8.10 Public Host Protocol Guidelines”, dated August 2000.¹

The following table details support for PHP protocol 8.10 commands/responses:

Command Description	Code	Notes
<i>Read Channel</i>	0x00	Channel address uses card number as two hex characters (0x00-0xFF) instead of single character defined by protocol doc (0x00-0x07)
<i>Write Channel</i>	0x01	Channel address uses card number as two hex characters (0x00-0xFF) instead of single character defined by protocol doc (0x00-0x07)
<i>Read Point</i>	0x02	Not Supported in AX driver
<i>Write Point</i>	0x03	Not Supported in AX driver
<i>Acknowledge Transaction</i>	0x05	Used for native alarm support in AX driver
<i>Read Next Channel</i>	0x06	Supports device learn in AX driver
<i>Read Next Point</i>	0x07	Not Supported in AX driver
<i>Read Storage Record</i>	0x08	Used for reading a file from panel to JACE
<i>Write Storage Record</i>	0x08	Used for writing a file from JACE to panel
<i>Read Value and Status</i>	0x0A	Not Supported in AX driver
<i>Read Message Text</i>	0x0B	Used for native alarm support in AX driver
<i>Set Time and Date</i>	0x0C	Used to synch panel date/time to JACE date/time if configured to do so, at transitions from off-line to on-line and by user command.
<i>Restart</i>	0x0D	Not Supported in AX driver.
<i>Exchange Virtual Terminal Packets</i>	0x0E	Used for Virtual Terminal support
<i>Say “Hello”</i>	0x0F	Used for ping and some device learn functions
<i>Open Storage File</i>	0x10	Used for panel/JACE file operations in AX driver
<i>Close Storage File</i>	0x11	Used for panel/JACE file operations in AX driver
<i>Initialize Name Cache</i>	0x12	Not Supported in AX driver.
<i>Fetch through Cache</i>	0x13	Not Supported in AX driver.

¹ “Php Version 8.10 Public Host Protocol”, August 2000, American Auto-Matrix, One Technology Lane, Export, Pennsylvania 15632

Information and specifications published here are current as of the date of publication of this document. Tridium, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia. Products or features contained herein may be covered by one or more U.S. or foreign patents.

Platforms

<i>Command Description</i>	<i>Code</i>	<i>Notes</i>
<i>Change Default Operator</i>	0x14	Not Supported in AX driver.
<i>Report Extended ID</i>	0x15	Not Supported in AX driver.
<i>Open Virtual Terminal</i>	0x16	Used for Virtual Terminal support
<i>Close Virtual Terminal</i>	0x17	Used for Virtual Terminal support
<i>Transport</i>	0x04	Not Supported in AX driver
<i>Transport (SF1 Only)</i>	0x19	Not Supported in AX driver
<i>PUP command pass-through</i>	0x18	Not Supported in AX driver

Platforms

The PHP driver will function on all AX platforms that support serial communications.

Installation

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

From your PC, use the Niagara Workbench 3.*n.nn* 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.

Apart from installing the 3.*n.nn* version of the Niagara distribution files in the JACE, make sure to install the **aaphp** 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 PHP software integration, as described in the rest of this document.

Quick Start

- Follow the installation and configuration instructions preceding this.
- Open the aaphp palette and find the AaPhpStarNetwork object.
- Paste an AaPhpStarNetwork object under the “drivers” node in your station.
 - Edit the Serial Port Config properties as needed.
- Open the PhpDeviceManager view by double-clicking the AaPhpStarNetwork object just added to the station.
 - Press the “Discover” button and provide a min and max unit number range to narrow the discovery search.
 - Select and add the PHP device you wish to integrate.
- Once a device is added, navigate to the “points” folder under the device and double click the “points” folder to display the PhpPointManager view.
 - Press the “Discover” button and provide channel/attribute search criteria.
 - Select and add PHP points (php attributes) to your database.

Php Component Guides

AaPhpStarNetwork

The AaPhpStarNetwork provides all the configuration parameters necessary to allow the driver to communicate with a network of PHP devices. The AaPhpStarNetwork can be thought of as the “host” PHP device in network of PHP devices.

The AaPhpStarNetwork 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 Users Guide for more information), as well as properties unique to operation as a master or peer in a PHP system.

Property	Value
Status	{ok}
Enabled	true
Fault Cause	
Health	Ok [01-May-06 4:42 PM EDT]
Alarm Source Info	Alarm Source Info
Monitor	Ping Monitor
Tuning Policies	Tuning Policy Map
Poll Scheduler	Basic Poll Scheduler
Retry Count	2
Response Timeout	+00000h 00m 02.000s
Inter Message Delay	00000h 00m 00.000s [0ms - 1sec]
Serial Port Config	Serial Helper
Terminal History Capacity	Record Count 1000 records
Device02	Aa Php Star Device

NOTE: In the following properties, the grayed out properties are inherited from the base AX driver classes, and as such are only touched on here. For a full explanation, refer to the AX Driver Architecture document.

- Status – The status of the network. Will normally be {ok}. A value of {fault} usually indicates an incorrect Serial Port Config setting.
- Enabled – Enables or Disables the Php Driver
- Fault Cause – if the “Status” is fault, the fault cause will appear here. For instance, if a non-existent COM port is configured into the “Serial Port Config” property, then this will say “Could not enable serial communication (javax.baja.serial.PortClosedException)”.
- Monitor – container for monitor (ping) properties.
- Tuning Policies - A container for tuning policies which determines how and when proxy points are read and written.

Information and specifications published here are current as of the date of publication of this document. Tridium, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia. Products or features contained herein may be covered by one or more U.S. or foreign patents.

- Poll Scheduler - The basic poll scheduler enables/disables polling, determines the fast/normal/slow poll rates, and maintains statistics about proxy extension polls.
- Retry Count – Determines how many retries the communications handler will try to re-send a message if the initial attempt is unsuccessful. For PHP protocol, this should normally be set to 1.
- Response Timeout - Specifies the maximum time to wait for a response to a PHP message is sent. If a response is not received before the timeout, the PHP message is resent up to “Retry Count” times, each of which will wait for this Response Timeout period.
- Inter Message Delay – The minimum amount of time to wait between receiving a message on the PHP bus, and sending the next request. This gives time for some PHP devices to prepare for receiving messages again. Note that this setting this value to non-zero has a negative impact on overall throughput, but may be necessary if a slow-to-turn-around PHP device is on the network.
- Serial Port Config – A container for the serial port properties.
 - Status - {ok} or {fault}
 - Port Name – Common name of a serial port, such as “COM1”
 - Baud Rate – Selected from a drop-down list.
 - Data Bits – select 5,6,7, or 8 bits – Use 8 bits for PHP protocol.
 - Stop Bits – select 1 or 2 bits – Use 1 stop bit for PHP protocol.
 - Parity – select None, Odd, Even, Mark, or Space – Use None for PHP protocol.
 - Flow Control Mode – Do not select any flow control for PHP protocol.
- Terminal History Capacity – The AaPhpStarNetwork is able to save a limited number of lines of terminal sessions. This property sets the max number of records that can be saved in the Terminal History database.

AaPhpStarDevice

The AaPhpStarDevice provides all the configuration parameters necessary to allow the driver to communicate with a given PHP Star device. The AaPhpStarDevices are always children of a AaPhpStarNetwork. AaPhpStarDevices also serve as a container for all of the data points that need to be polled for data (a PHP attribute under a channel in a device)

The AaPhpStarDevice is the "device-level" component in the NiagaraAX architecture.

Device02 (Aa Php Star Device)	
Status	{unackedAlarm}
Enabled	<input checked="" type="checkbox"/> true
Fault Cause	
Health	Ok [01-May-06 4:57 PM EDT]
Alarm Source Info	Alarm Source Info
Address	2 [0 - 31]
Points	Aa Php Star Point Device Ext
Aa Php Alarm Info	Aa Php Native Events Processor
Terminal Mode Inter Message Delay	+00000h 00m 00.200s
Allow Time Sync	<input checked="" type="checkbox"/> true

NOTE: In the following properties, the grayed out properties are inherited from the base AX driver classes, and as such are only touched on here. For a full explanation, refer to the AX Driver Architecture document.

- 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 “Report Exception Message” is used as the ping message).
- Enabled – Enables or Disables communication to the associated device from Php Driver
- Fault Cause – if the “Status” property value is {fault}, the fault cause is displayed here.
- Health – 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 comm. Failure (example – “bad checksum)
- Alarm Source Info – configuration items for alarms generated from the ping process (device up and device down events)
- Address – the address of the target device (the “unit number”)

Information and specifications published here are current as of the date of publication of this document. Tridium, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia. Products or features contained herein may be covered by one or more U.S. or foreign patents.

AaPhpStarDevice

- **Points** – a container for all data items (attributes) in this device which need to be polled for data.
- **Aa Php Alarm Info** – configuration data to apply to alarms that originate in PHP devices and are retrieved using the “Read Message Text” message. PHP native alarms are normalized and passed to the AX alarming subsystems which then display and manage them. Acknowledgments are in turn transmitted back down to the PHP device that originated them.
- **Terminal Mode Inter Message Delay** – When in terminal mode, terminal packets are transmitted inter-weaved with other “normal” communications, such as polls for data and pings for device status. In order to keep the terminal session from consuming the entire bandwidth of the communications bus, a delay between terminal mode messages is introduced. This allows other communications to take place. To assign less bandwidth to the terminal session and more to the other communication messages, increase this value. Default is 200ms, which allows about a 1:1 interleave of terminal messages and other messages.
- **Allow Time Sync** – allow a time sync to be transmitted to this device. This time sync is invoked (if enabled by this property) from a action on the device (right-click the AaPhpStarDevice, select “Actions”). Time sync is also invoked (if enabled by this property) whenever this device’s Status property has been {down} and transitions to {ok}.

AaPhpStarProxyExt

The AaPhpStarProxyExt type is a proxy for one piece of data of interest in a PHP controller. This single piece of data is defined by a combination of the channel and the attribute in the channel.

The AaPhpStarProxyExt type takes on the readable-writable personality of the control point they are attached to. For example, an AaPhpStarProxyExt, 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 attribute values.

The AaPhpStarProxyExt is the "point-level" component in the NiagaraAX architecture.

Property	Value
Status	{ok}
Fault Cause	
Enabled	true
Device Facets	>> [Refresh]
Conversion	Default
Tuning Policy Name	Default Policy
Read Value	2.5 {ok}
Write Value	0.0 {ok}
Aa Php Ord	Star SU=02 T=5y Ca=00 Ch=00 At=VR
Poll Frequency	Normal

NOTE: In the following properties, the grayed out properties are inherited from the base AX driver classes, and as such are only touched on here. For a full explanation, refer to the AX Driver Architecture document.

- 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 “Report Exception Message” is used as the ping message).
- Fault Cause – if the “Status” property value is {fault}, the fault cause is displayed here.
- Enabled – Enables or Disables communication to the associated device from Php Driver
- Device Facets – contains metadata about the health of this device on the network:
- Conversion
- Tuning Policy Name
- Read Value
- Write Value

Information and specifications published here are current as of the date of publication of this document. Tridium, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia. Products or features contained herein may be covered by one or more U.S. or foreign patents.

- Aa Php Ord – a special collection of data which uniquely defines an atomic data item within a controller. Consists of various subfields. These numbers are typically determined automatically during a device learn, but can be manually entered if necessary:

<input checked="" type="radio"/> Aa Php Ord	Star SU=02 T=Sy Ca=00 Ch=00 At=VR	
<input type="checkbox"/> Sub Unit	<input type="text" value="2"/>	[0 - 255]
<input type="checkbox"/> Card	<input type="text" value="0"/>	[0 - 255]
<input type="checkbox"/> Channel	<input type="text" value="0"/>	[0 - 255]
<input type="checkbox"/> Fundamental Type	<input type="text" value="Sy"/>	▼
<input type="checkbox"/> Attr	<input type="text" value="VR"/>	
<input type="checkbox"/> Numeric Encoding Type	<input type="text" value="1"/>	Radix=16 [0 - ff]

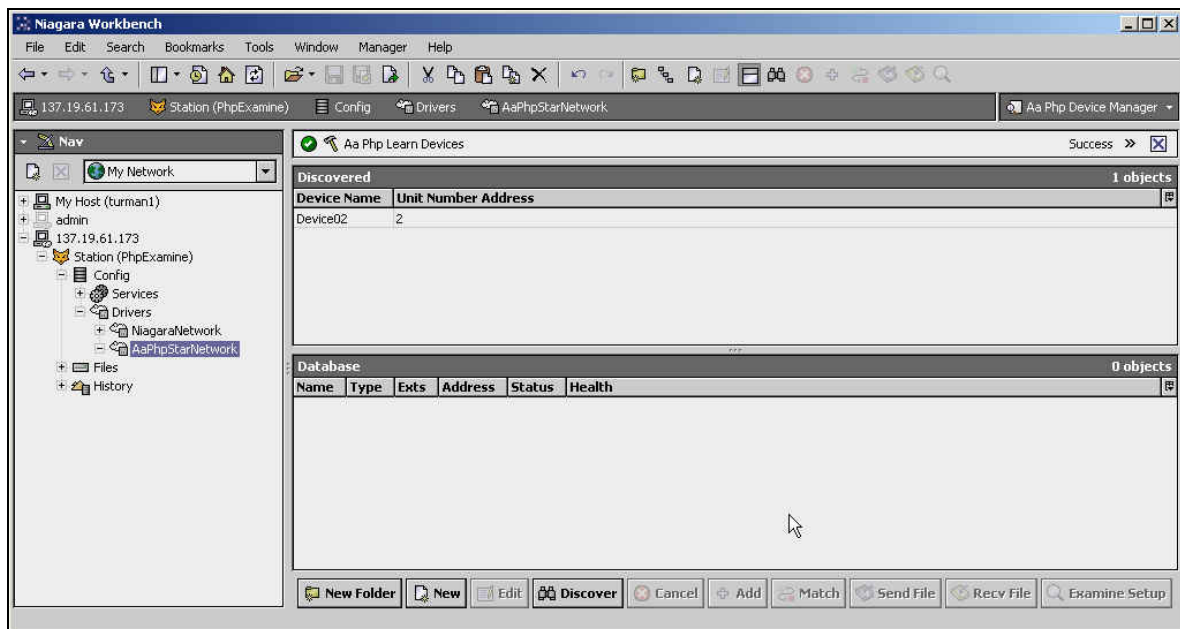
- Sub Unit: Typically (but does not have to be) the same as the unit number on a Star device.
- Card: The card within a Star device (sub unit).
- Channel: The hardware channel on the card on which the data item resides.
- Fundamental Type: one of several types as reported by the device
- Attr: The two-letter mnemonic for the attribute.
- Numeric Encoding Type: Read only value: An encoding type reported in the data responses. Tells Niagara how to interpret the value.
- Poll Frequency – the frequency to poll this point in the controller. Choices are either fast/normal/slow. Fast/Normal/Slow rates are determined by the “Poll Scheduler” property on the PhpNetwork component.

Php Views

Aa Php Star Device Manager

The Php Star Device Manager is the default view when you double-click on a Php Network in the Nav tree. This manager view provides a quick and easy way to display and learn PHP devices that are on the PHP network:

The Php Device Manager is the default view for any Php Star Network container. The Php Star Device Manager is a table-based view, where each row represents a unique device. When building a network in the station, you use this view to create, edit, and delete device-level components. Below is an example Php Star Device Manager view.



The Php Device Manager consists of two main panes. The view above shows an example Php Star Device Manager view.

The “New Folder”, “New”, and “Edit” buttons are not unique to the Php Device Manager, and are explained in the “Niagara AX User’s Guide” in the “Driver Architecture” section. The “Match” button is not used for the PHP driver.

The “Discover” button does implement functionality that is unique and tailored to discovering PHP Star devices. By clicking the “Discover” button, the “learn” mode of the manager is invoked, and a dialog box pops up to prompt the user for some additional data to guide the discovery process as shown in the following figure:

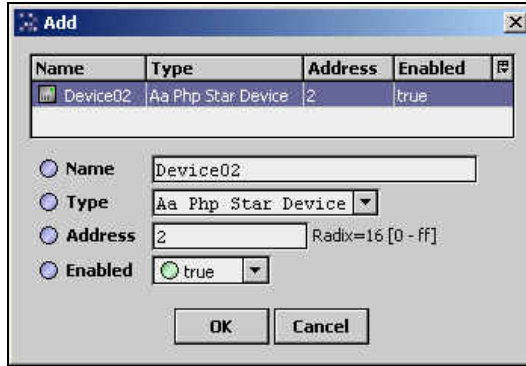


This dialog box contains four data items:

- **Send Who Are You:** Defaults to true. If true, the learn sends out a “?” command. All Star panels on the network respond to the “?” command with its own unit number. If a Star panel is known to be the only panel on a network, then it is appropriate to leave this set to “true”. However, if multiple Star panels are on the network, all panels will attempt to answer the “?” command with its own address at the same time, resulting in an incorrect address being reported. ***If multiple star panels are alive on the network at the time of device discovery, it is recommended that this parameter be changed to “false”. Otherwise, incorrect device addresses may be learned.*** If there is only one Star panel online at the time of a device discovery and this parameter is set to true, then the “From” and “To” properties that follow are of no consequence and can be ignored.
- **From:** The starting unit number to discover. This parameter is used if “Send Who Are You” is set to false, or if there is no response to the “Who Are You” (“?”) command.
- **To:** The ending unit number to discover. This parameter is used if “Send Who Are You” is set to false, or if there is no response to the “Who Are You” (“?”) command.
- **Timeout:** The amount of time to wait for a response from a device before discovering the next device. If the device does not answer within this time, it assumed to not be present, and will not be added to the discovery list. Note that no retries are used during the discovery process.

The progress of the discover devices process can be viewed along the “progress bar” at the top of the manager view. In addition, a log of addresses searched, and the search results, can be viewed by clicking the small “>>” button to the immediate right of the job bar.

Once the discovery job is complete, the top half-pane of the point manager will display a table of devices discovered. If you highlight one or more rows in the top “Discovered” pane, the “Add” button becomes active. You can now add the selected devices to the station database by clicking the “Add” button. This pops up the “Add” dialog box, as shown in the following figure:



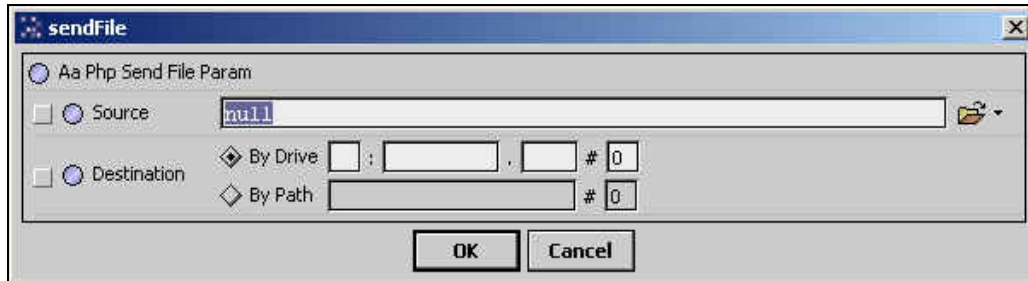
The “Add” dialog box affords you the opportunity to tweak the display name, enabled state, and/or address of each of the selected devices. Click the “OK” button to add the devices to the database, or click “Cancel” to bail out.

“Send File”, “Receive File”, and “Examine Setup”

If a user clicks on a device in the lower pane, the “Send File”, “Receive File”, and “Examine Setup” buttons become active:



- The “Send File” button is used to send files from the JACE or Workstation down to the Star Panel. The parameters for the send need to be set correctly by the user for the desired download.



Source: Click the file chooser button on the far right to display a file chooser dialog and select the file to be downloaded. The corresponding file Ord will be displayed. Files on the JACE or on the client (Workbench) side may be selected. The last used source will be displayed by default.

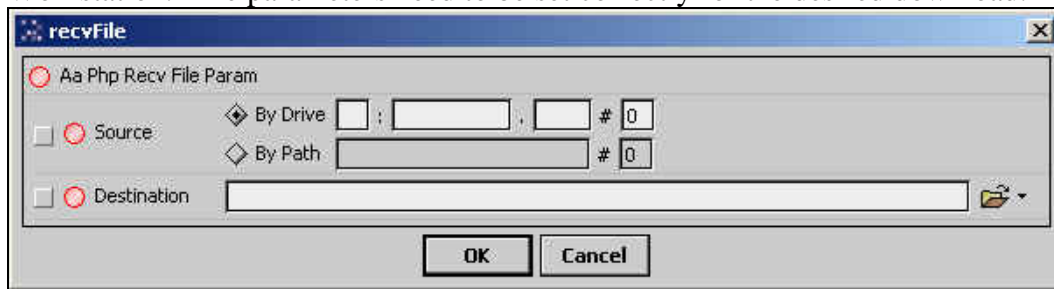
Destination: The destination for Star Panels should use the “By Drive” selection. The “By Drive” driver number is typically the same as the device’s unit number. The filename and extension should be the well-known names and extensions for PHP Star panels, such as “SYSTEM.SYS” or “DATABASE.SYS” typically. Other known names are “VARIABLE.SYS”, trendname.TRN (where trendname is 8 characters or less), and MODULE*n*.IOM (where *n* is a number of a C module). The instance number (following the “#” sign) should typically be left at zero.

Information and specifications published here are current as of the date of publication of this document. Tridium, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia. Products or features contained herein may be covered by one or more U.S. or foreign patents.

For example, to download the SYSTEM.SYS file to unit #2 from the file “system.sys” on the JACE:



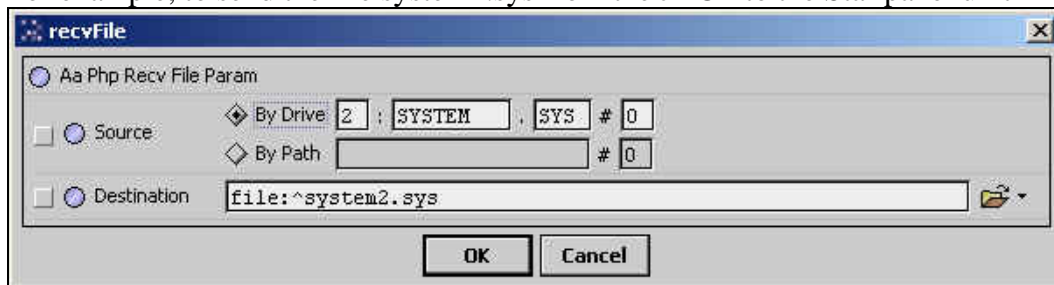
- The “Receive File” button is used to send files from the Star Panel to a JACE or Workstation. The parameters need to be set correctly for the desired download.



Source: The source for Star Panels should use the “By Drive” selection. The “By Drive” driver number is typically the same as the device’s unit number. The filename and extension should be the well-known names and extensions for PHP Star panels, such as “SYSTEM.SYS” or “DATABASE.SYS” typically. Other known names are “VARIABLE.SYS”, trendname.TRN (where trendname is 8 characters or less), and MODULE n .IOM (where n is a number of a C module). The instance number (following the “#” sign) should typically be left at zero.

Destination: Click the file chooser button on the far right to display a file chooser dialog and select the destination file name. The corresponding file Ord will be displayed. Files on the JACE or on the client (Workbench) side may be selected. If the file does not exist, it will be created. If the file does exist, it will be overwritten.

For example, to send the file system2.sys from the JACE to the Star panel unit #2:



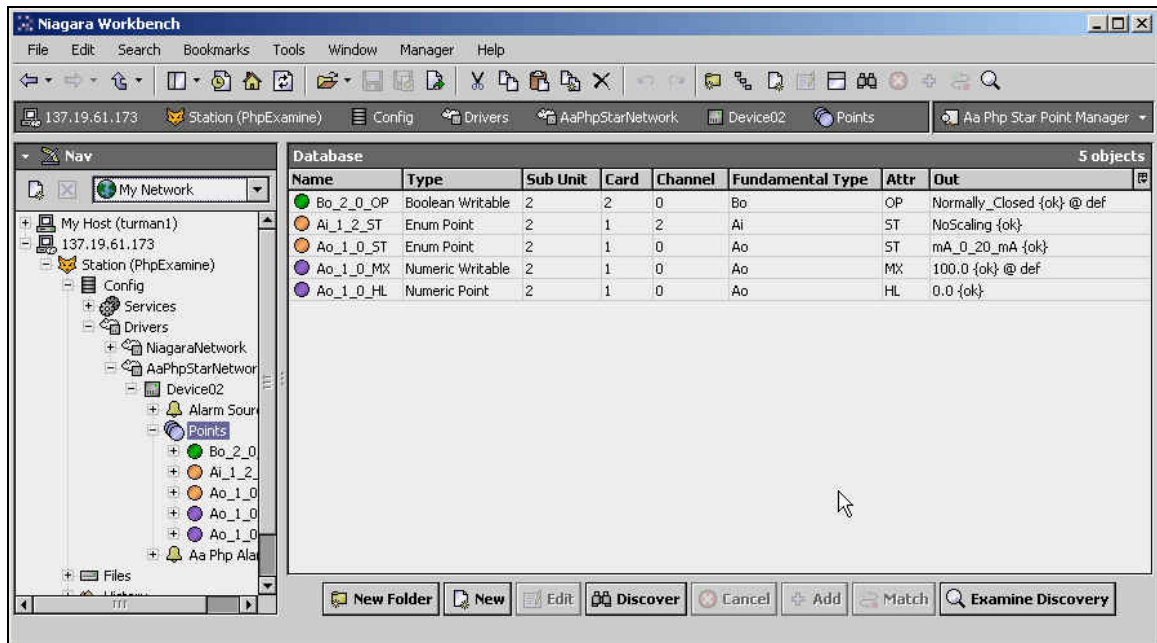
- Examine Setup:** This button is used to create and start a job that is used to detect changes to the panel’s internal setup. See appendix A for operation.

Information and specifications published here are current as of the date of publication of this document. Tridium, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia. Products or features contained herein may be covered by one or more U.S. or foreign patents.

Php Star Point Manager

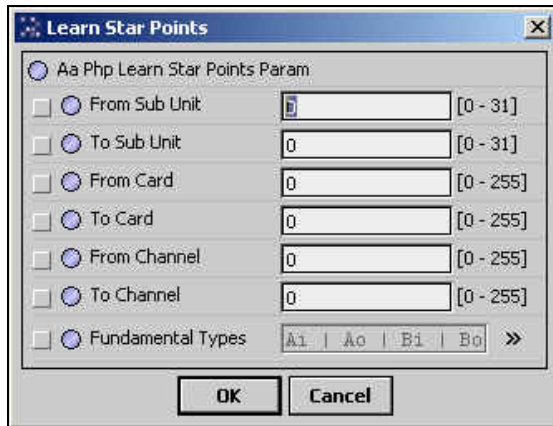
The Php Star Point Manager is the default view when you double-click on a “points” folder (a AaPhpStarPointDeviceExt type folder) under an AaPhpStarDevice in the Nav tree. This manager view provides a quick and easy way to display and learn PHP points in a PHP Star device.

The Php Star Point Manager is the default view for any AaPhpStarPointDeviceExt container. The Php Star Point Manager is a table-based view, where each row represents a unique PHP attribute within a device. Below is an example Php Star Point Manager view.



The “New Folder”, “New”, and “Edit” buttons are not unique to the Php Device Manager, and are explained in the “Niagara AX User’s Guide” in the “Driver Architecture” section. The “Match” button is not used for the PHP driver.

The “Discover” button implements functionality that is unique and tailored to discovering PHP points in a Star panel. 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 in the following figure:



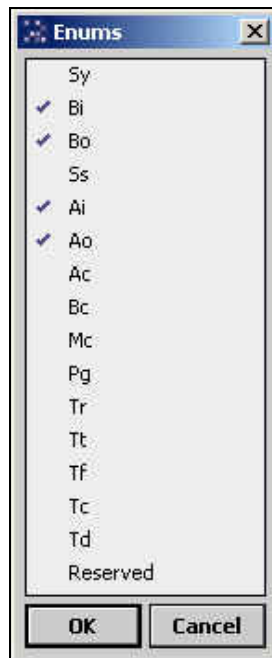
The range of discovery can be customized by entering in appropriate entries for each of the above:

“From Sub Unit” and “To Sub Unit”: typical values are 0 and 9.

“From Card” and “To Card”: typical values are 0 and 9

“From Channel” and “To Channel”: typical values are 0 and 9

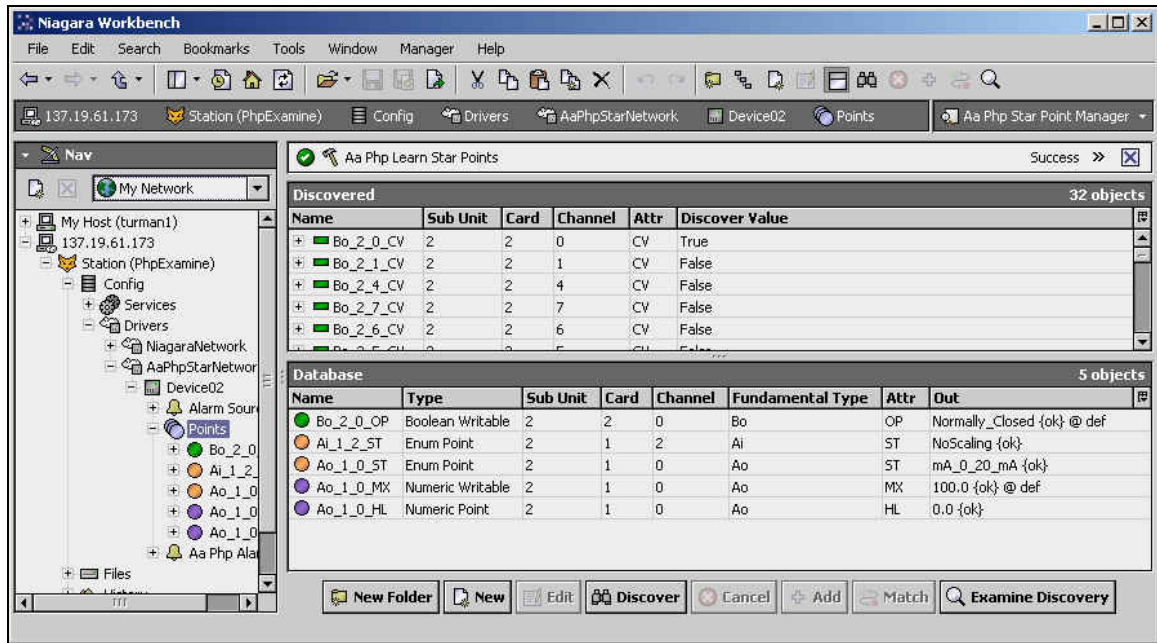
“Fundamental Types”: Click on the “>>” to bring up a enumeration list of fundamental types to learn and then click OK to close the dialog box:



When you click OK, the channels will be searched for all attributes, and the results will be displayed in the top pane of the following split-pane format, as shown in the following figure:

Information and specifications published here are current as of the date of publication of this document. Tridium, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia. Products or features contained herein may be covered by one or more U.S. or foreign patents.

Php Star Point Manager



Note that the first (default) attribute of each channel is listed, with a “+” mark in front of the channel number if there are additional attributes in that channel. To access the additional attributes, click on the “+” sign to expand:

+ Bo_2_0_CV	2	2	0	CV	True
+ Bo_2_1_CV	2	2	1	CV	False
Bo_2_1_OP	2	2	1	OP	Normally_Closed
Bo_2_1_AM	2	2	1	AM	Manual
Bo_2_1_RL	2	2	1	RL	0 hr
Bo_2_1_RH	2	2	1	RH	0 hr
Bo_2_1_RM	2	2	1	RM	0 min
+ Bo_2_4_CV	2	2	4	CV	False

Single or multiple attributes can be added as control points with AaPhpStarProxyExt extensions by selecting the discovered row(s) in the top pane, and clicking add. Doing so will cause the “Add” dialog box to appear, as shown in the following figure:

Name	Type	Sub Unit	Card	Channel	Fundamental Type	Attr	Numeric Encoding Type	Enabled	Facets
Bo_2_1_CV	Boolean Point	2	2	1	Bo	CV	a	true	units=null, precis
Bo_2_1_OP	Boolean Point	2	2	1	Bo	OP	d	true	units=null, precis
Bo_2_1_AM	Boolean Point	2	2	1	Bo	AM	b	true	units=null, precis
Bo_2_1_RL	Numeric Point	2	2	1	Bo	RL	6	true	units=hr, precis
Bo_2_1_RH	Numeric Point	2	2	1	Bo	RH	6	true	units=hr, precis
Bo_2_1_RM	Numeric Point	2	2	1	Bo	RM	5	true	units=min, precis

Name: Bo_2_1_CV
 Type: Boolean Point
 Sub Unit: 2 [0 - 255]
 Card: 2 [0 - 255]
 Channel: 1 [0 - 255]
 Fundamental Type: Bo
 Attr: CV
 Numeric Encoding Type: a Radix=16 [0 - ff]
 Enabled: true
 Facets: units=null, precision=0, min=0.0, max=1.0, trueText=True, falseText=False
 Tuning Policy Name: Default Policy
 Device Facets: >>
 Conversion: Default

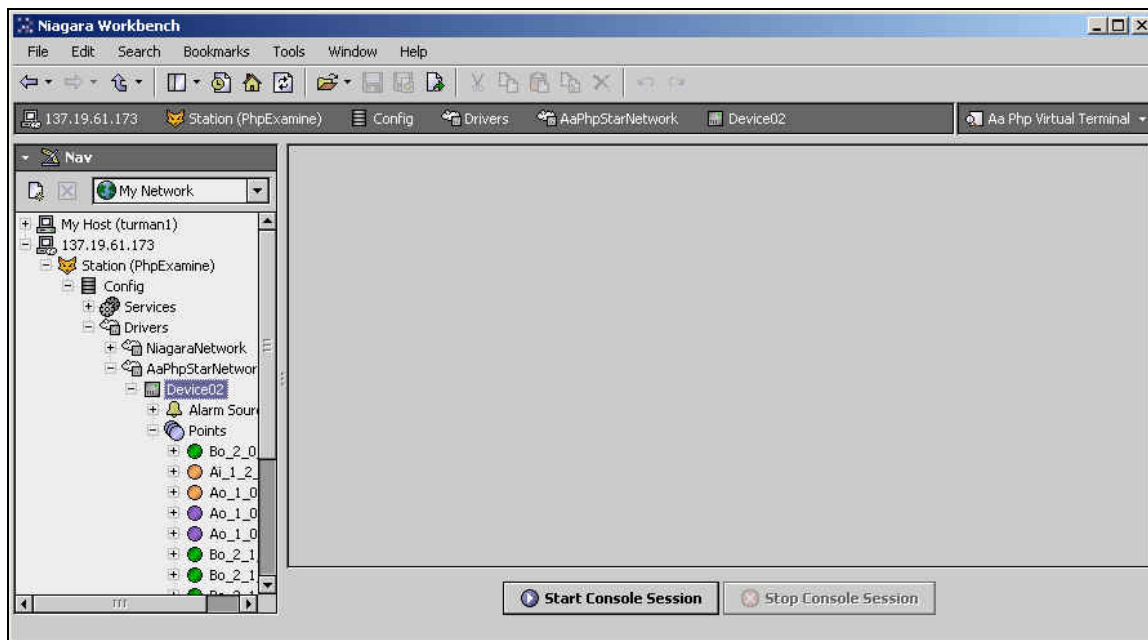
OK Cancel

In this dialog, the points may be edited individually or in a batch. The point “Type” defaults to the appropriate “read-only” version of a control point. If a writable version is desired, be sure to change it now before adding it to the database.

Once the point(s) are satisfactorily edited, click “OK” to create the proxy points corresponding to the channel attributes.

Php Virtual Terminal View

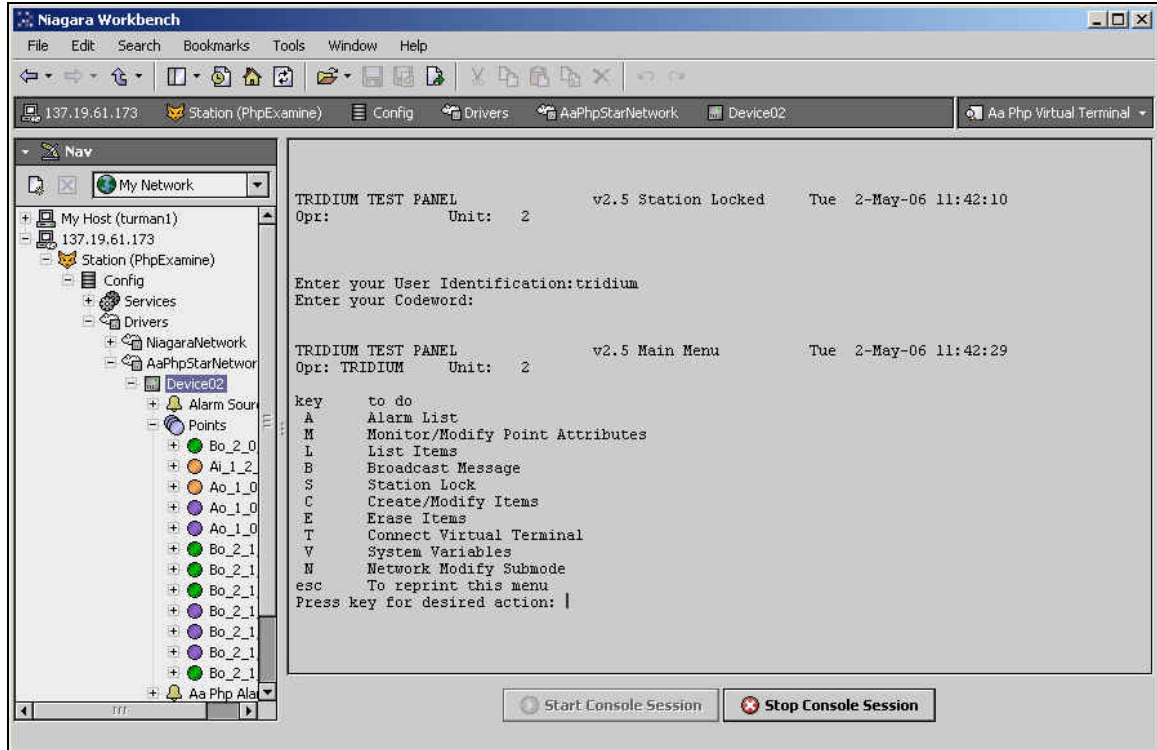
The Php Virtual Terminal View is the default view when you double-click on a “device” (a AaPhpStarDevice type) in the Nav tree. This manager view provides a way to view and modify the configuration within a PHP panel:



The virtual terminal allows a Niagara operator to access the Star panel just as if they were attached locally with a terminal to the Star panel. The view provides a single large display pane and two buttons to start and stop the console session. A console session is started by clicking the “Start Console Session” button. The JACE will then send the commands to the PHP Star panel to open a virtual terminal session. The “Stop Console Session” will become enabled if the open virtual terminal commands were successful. It will route any text received from the session back to the Virtual Terminal view pane. All keystrokes typed into the session will be routed to the PHP Star panel. The first text sent back from the Star panel after opening a virtual terminal session is the “User Identification” and “Codeword” prompts.

An example session is shown in the following figure:

Php Virtual Terminal View



There are two ways to stop a console session. The most obvious is to click on “Stop Console Session”, which will then cause the stop virtual terminal command to be sent to the Star panel.

The second way to stop a console session is to navigate away from the console view. Any time a user navigates away from the console view while a virtual terminal session is active will cause the virtual terminal session to terminate.

Php “CV” Attributes on Numeric Points

Whenever a PHP proxy extension is added or configured to have an Attribute property value of “CV” and the base Type is “Numeric Point”, as shown below, there is additional special behavior attached to the point:

Adding a Numeric Point with CV Attribute

Name	Type	Sub Unit	Card	Channel	Fundamental Type	Attr	Numeric Encoding Type
Ai_1_0_CV1	Numeric Point	2	1	0	Ai	CV	1c

The dialog box shows the following configuration details:

- Name:** Ai_1_0_CV1
- Type:** Numeric Point
- Sub Unit:** 2 [0 - 255]
- Card:** 1 [0 - 255]
- Channel:** 0 [0 - 255]
- Fundamental Type:** Ai
- Attr:** CV
- Numeric Encoding Type:** 1c Radix=16 [0 - ff]
- Enabled:** true
- Facets:** units=null,precision=1,min=-3276.8,max=3276.7
- Tuning Policy Name:** Default Policy
- Device Facets:** >>
- Conversion:** Default

In this case, the additional behavior is the addition of a command to set an override value to the point. However, this behavior also requires some additional configuration to perform correctly. In particular, the “prewriteAttribute” property of the proxy extension must be set to the appropriate value, as shown in the following figure:

The 'Prewrite Attribute' dropdown menu is shown with the following options:

- None
- Oi
- Am

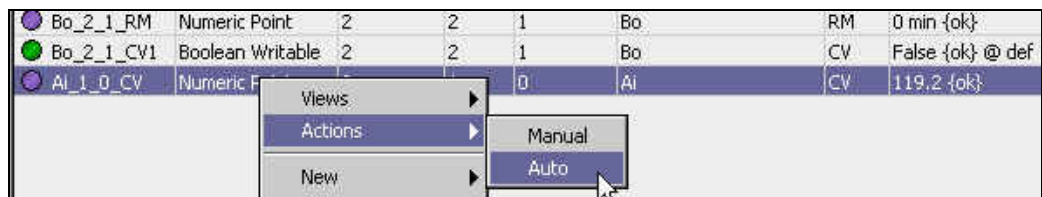
This property is invisible until an attribute value of “CV” is entered, whereupon its visibility is changed. This value defaults to “None”. If it is changed to “Oi” or to “Am”, then a dynamic action is added to the proxy point to allow setting of the value CV.

Here are the basic rules to be able to use the set action:

- 1) The point type must be Numeric Point (Note: Numeric Writable does not work)
- 2) The “Attribute” property must be set to “CV”. This will make the “Prewrite Attribute” property visible.
- 3) The “Prewrite Attribute” property must be set to “Oi” or “Am” depending on whether the OI attribute or the AM attribute must be written before the CV value can be written. This varies from channel to channel, device to device.
- 4) If the “Prewrite Attribute” is “Oi” or “Am”, then an appropriate pair of actions is added to the proxy extension.
 - a. If the Prewrite Attribute value is set to “Oi” then the actions **Override** and **CancelOverride** are added to the proxy extension as shown.



- i. When the **Override** action is invoked, it will set the attribute “OI” in the controller to 1, followed by a write of the user entered value to the attribute “CV”.
 - ii. When the **CancelOverride** action is invoked, it will reset the attribute “OI” in the controller to 0.
- b. If the Prewrite Attribute value is set to “Am” then the actions **Manual** and **Auto** are added to the proxy extension as shown.



- i. When the **Manual** action is invoked, it will set the attribute “AM” in the controller to 0, followed by a write of the user entered value to the attribute “CV”.
 - ii. When the **Auto** action is invoked, it will reset the attribute “AM” in the controller to 1.

Examine Setup and Examine Discovery Features

Two special features of the driver were added in version 3.0.100.1 to document the configuration of a Star panel and to determine if the configuration has changed:

- 1) Examine Setup
- 2) Examine Discovery

The combined usage of these two features allows for complete documentation of a baseline, or initial configuration, of a Star panel at a point in time, and allows for a comparison in the future to document/discover changes to the programs configuration. Note that usage of Examine Setup or Examine Discovery alone does not supply complete documentation – both are required for a complete document.

The “Examine Setup” feature is accessed via the Aa Php Device Manager. The Aa Php Device Manager is the default screen that you see when you double click the Aa Php Star Network. On the Aa Php Device Manager, you will see a button, toolbar item, and menu item with a magnifying glass icon named “Examine Setup”. You will use this new feature to determine if there are any major changes to the setup in the star panel.

The “Examine Discovery” is accessed via the Aa Php Point Manager view, accessed when you double click on the “points” folder under a device. On the Point Manager, you will see a button, toolbar item, and menu item with a magnifying glass icon named “Examine Discovery”. You will use this new feature to determine if there are any major changes to the point configurations in the star panel.

Examine Setup

Examine Setup Script

The “Examine Setup” button uses a special .xml script file to automatically send virtual terminal commands to the Star panel and record the responses in a log file. This file (or files) must be created before the Examine Setup feature is used.

There are 4 XML tags, described as follows. Note that the tags and attributes *are case-sensitive*:

<TerminalCommands>...</TerminalCommands>

This is the root tag of the xml file. Exactly one set of these tags should be present. The other tags mentioned below should be inside of this set of tags.

<KeyStrokes></KeyStrokes>

Any text between these tags will be sent to the terminal as if a user were typing on the keyboard. Special characters are escaped using standard xml escape sequences.

- **NOTE:** This tag has an optional attribute “Repeat”. If present, the value should be an integer. The text in this tag will be sent to the terminal once if no “Repeat” attribute is supplied, otherwise, the text will be sent as many times as specified by the “Repeat” attribute.

<Sleep></Sleep>

The text between these tags should specify an interval in milliseconds. When encountered, the process that evaluates the xml file will pause for this number of milliseconds. This is included to allow the Star panel time to send responses back.

<Gather>...</Gather>

Any combination of <Sleep>..</Sleep> or <KeyStrokes>..</KeyStrokes> may be included inside a set of <Gather>..</Gather> tags. The results of any tags that are inside a set of <Gather>..</Gather> tags will be saved to a results file on your hard-drive. The results file is described in step 2.

An example script file is shown here:

```
<TerminalCommands>
  <Sleep>1000</Sleep>
  <KeyStrokes>TRIDIUM&#x0D;</KeyStrokes>
  <KeyStrokes>NIAGARA&#x0D;</KeyStrokes>
  <Sleep>1000</Sleep>
  <Gather>
    <KeyStrokes>L</KeyStrokes>
    <KeyStrokes>E</KeyStrokes>
    <KeyStrokes repeat="50">&#x20;</KeyStrokes>
    <KeyStrokes>U</KeyStrokes>
    <KeyStrokes repeat="50">&#x20;</KeyStrokes>
    <KeyStrokes>M&#x0D;</KeyStrokes>
    <KeyStrokes repeat="50">&#x20;</KeyStrokes>
    <KeyStrokes>A&#x0D;</KeyStrokes>
    <KeyStrokes repeat="50">&#x20;</KeyStrokes>
    <KeyStrokes>G</KeyStrokes>
    <KeyStrokes repeat="50">&#x20;</KeyStrokes>
    <KeyStrokes>L</KeyStrokes>
    <KeyStrokes repeat="50">&#x20;</KeyStrokes>
    <KeyStrokes>&#x1B;</KeyStrokes>
    <KeyStrokes>V</KeyStrokes>
    <KeyStrokes>E</KeyStrokes>
    <KeyStrokes>&#x1B;</KeyStrokes>
    <KeyStrokes>V&#x0D;</KeyStrokes>
    <KeyStrokes repeat="100">N</KeyStrokes>
    <KeyStrokes>&#x1B;</KeyStrokes>
    <KeyStrokes>P</KeyStrokes>
    <KeyStrokes repeat="3">&#x1B;</KeyStrokes>
  </Gather>
</TerminalCommands>
```

NOTE: Creation of a script file requires special knowledge of the valid commands and sequences required to navigate the virtual terminal menus of a star panel, as well as how to encode special characters such as carriage-return, line-feed, and escape in XML.

Here are a few pointers:

- o Space is encoded

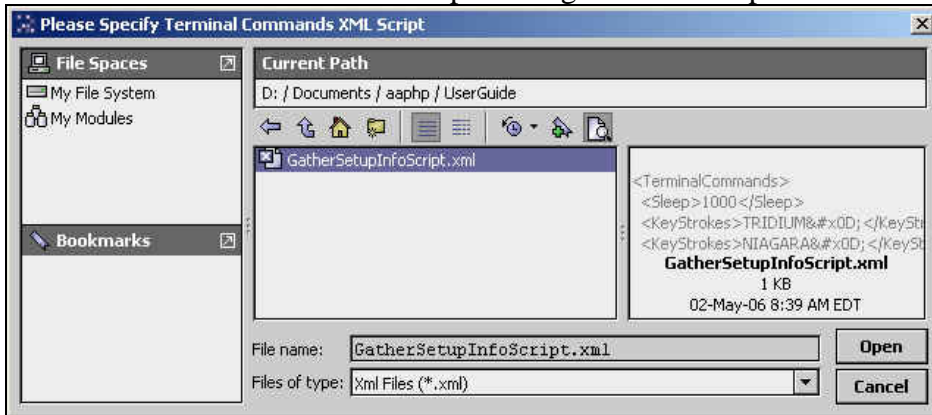
Information and specifications published here are current as of the date of publication of this document. Tridium, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia. Products or features contained herein may be covered by one or more U.S. or foreign patents.

- Carriage Return (or CR) is encoded
- Escape (or Esc) is encoded
- Control-A (ctrl-A) is encoded.....
- Control-N (ctrl-N) is encoded.....

Examine Setup Usage

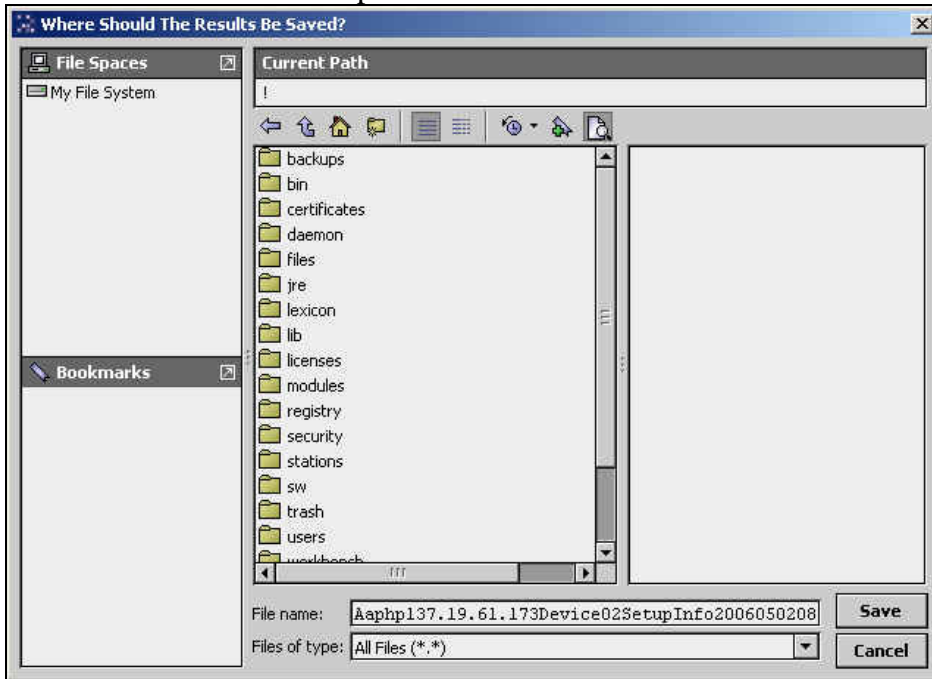
Step 1:

Once you have the XML script defined, as described in the previous section, you can launch the Examine Setup feature by clicking the Examine Setup button on the AaPhpDeviceManager view. When you do this, the file chooser dialog will display to allow selection of the examine script. Navigate to the script file and click **Open**:



Step 2:

Once you click Open above, you will next be prompted to select a location to save the results of the Examine Setup session:



Information and specifications published here are current as of the date of publication of this document. Tridium, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia. Products or features contained herein may be covered by one or more U.S. or foreign patents.

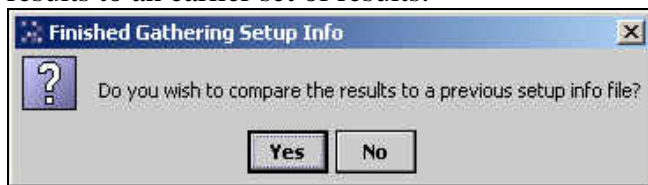
Navigate to the location you wish to save the results and select Save. Note that the default file name includes a device description and an encoded timestamp, but you may change the file name to any name that makes sense to your record keeping.

Step 3:

After step 2, the station will perform a virtual terminal session in the background, sending the commands from the XML file to the virtual terminal and saving the results to the specified location on your hard-drive. The device manager's job bar will show the progress for this step.

Step 4:

After step 3 completes, you will be asked whether or not you'd like to compare these results to an earlier set of results:

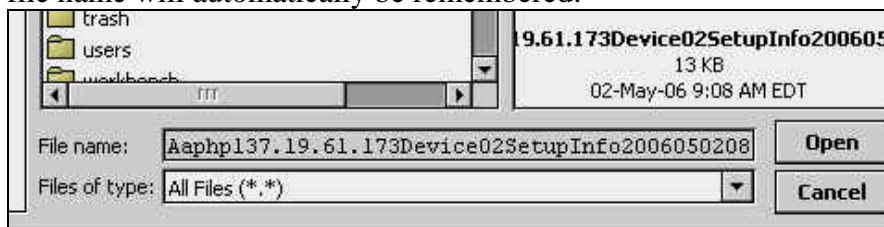


If this is your first time through, you might as well click “No” since there will be no earlier results. You now have a record of the panel configuration for future comparison. The results are text, so they are viewable using any standard text editor.

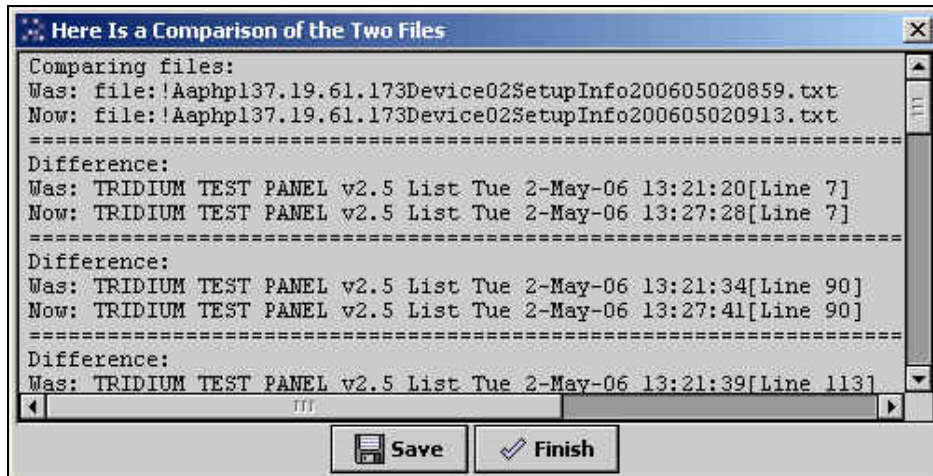
If you wish to compare the results with an earlier saved file, you should click “Yes” and continue on with the next steps.

Step 5:

Next you will be presented with another file chooser for the location of the results file from a previous setup examination session. If you are using the same computer then the previous file name will automatically be remembered:



When you click Open, the new and previous versions will be compared and a comparison results window will appear showing a simple text comparison of the two files, as shown in the following figure:

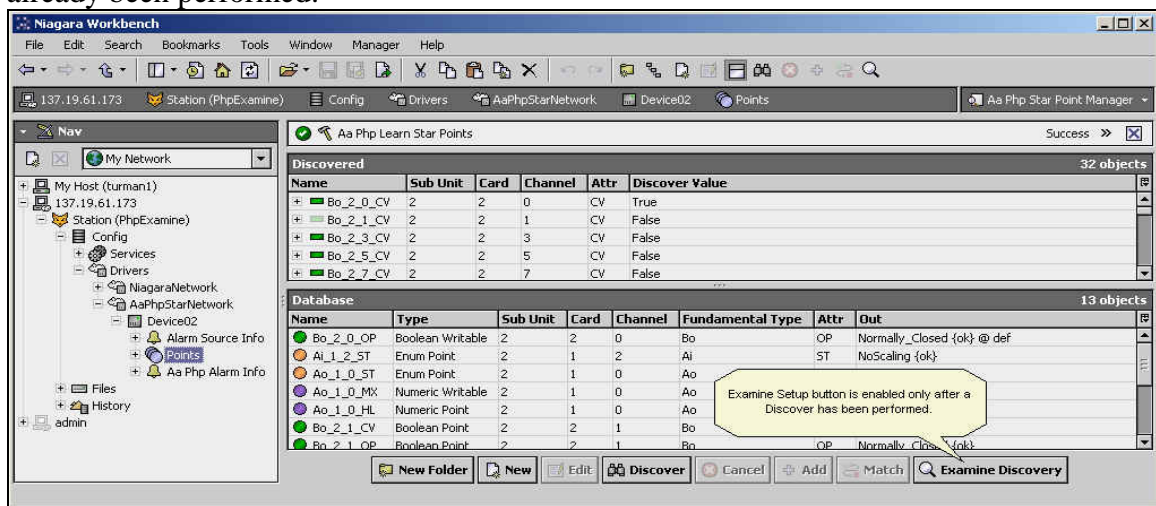


You may examine the results. The files are compared line by line, and any differences are presented. Most of the time the differences are a simple timestamp difference and may be safely ignored. The user will need to review the results for any significant, expected, or unexpected differences.

NOTE: This window has a “Save” button at the bottom to allow you to save what you see in the window to a text file. Clicking the “Save” button is optional. Click the “Finish” button to end the procedure.

Examine Discovery

The second of the new features is located on the Aa Php Star Point Manager. The Aa Php Star Point Manager is the default screen that you see when you double click the point-device-extension (named “Points” under any Aa Php Star Device). On the Aa Php Star Point Manager, you will see a new button, toolbar item, and menu item with a magnifying glass icon named “Examine Discovery”. This button is only active if a point discovery has already been performed:



Information and specifications published here are current as of the date of publication of this document. Tridium, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia. Products or features contained herein may be covered by one or more U.S. or foreign patents.

You will use this new feature to determine if there are any major changes to the list of available points and attributes in the star panel. The “Examine Discovery” button is grayed out until a point discovery has occurred.

Step 1:

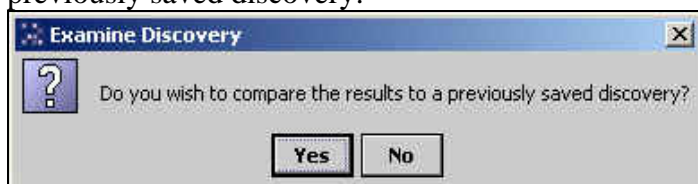
Perform a point discovery

Step 2:

Click the “Examine Discovery” button. This will cause a file chooser to display. Select a file name in which to save the results. Specify a location to save the information that you see in the “Discovered” pane. The information in the “Discovered” pane will be saved in an xml file on your hard-drive with an extension of “.bog”.

**Step 3:**

Next, you will be asked whether or not you'd like to compare this discovery to a previously saved discovery:



If this is your first time through, you might as well click “No” since there will be no earlier results. The results are now saved for a future comparison.

Otherwise, you should click “Yes” and continue with the next steps.

Step 4:

Next you will be presented with a file chooser dialog and asked for the location of the .bog file from the previous discovery. If you are using the same computer then the previous file name will automatically be remembered:

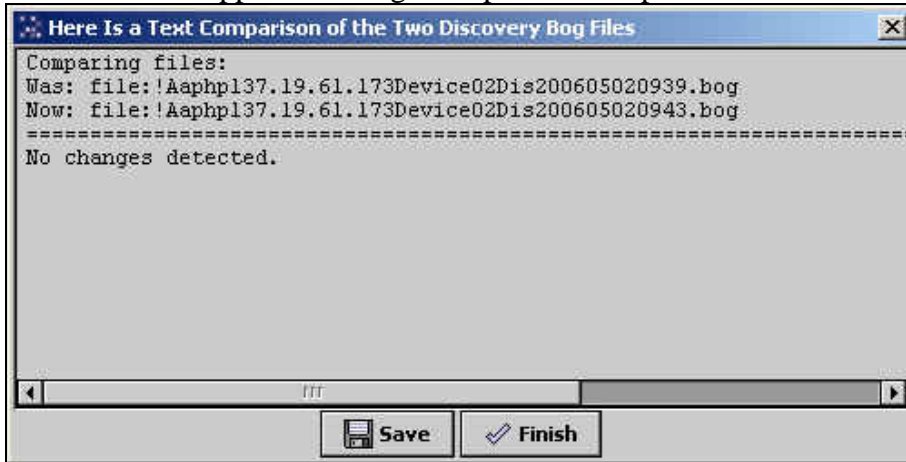


Select the desired bog file save from a previous discovery, and click Open.

Information and specifications published here are current as of the date of publication of this document. Tridium, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia. Products or features contained herein may be covered by one or more U.S. or foreign patents.

Step 5:

A window will appear showing a simple text comparison of the two files.



This window has a “Save” button at the bottom of it to allow you to save what you see in the window to a text file. Clicking the “Save” button is optional. Click the “Finish” button to end the procedure.