



## SnapAV WattBox Integration Protocol Document

Integration Protocol v3.0 rev20230330

### Overview

This integration protocol details how a third-party system can be used to control a SnapAV WattBox. With the WattBox online, the integration protocol will be listening for connections on port 23 or SSH on port 22 at the controller's IP address. To get started, netcat or similar software can be used to initiate a connection and test any of the following protocol commands below.

#### Important information:

The SSH feature was added with firmware 1.3.0.4. To connect with SSH you will need at least firmware version 1.3.0.4.

Only 10 simultaneous connections can be made at a time.

### Authentication

The protocol requires authentication before proceeding with commands. Once connected, a login prompt will be received, and the third-party system must provide a valid username and password. If correct, login will be successful and other commands can be issued. If incorrect, the third-party system will be prompted for login again.

#### Important information:

If using SSH, you will need to set a new password for the WattBox. There is a 13-character limit on passwords used for SSH user credentials.

### Specification

THIRD-PARTY SYSTEM <-----> SnapAV WattBox  
i.e. Wattbox IP: 192.168.0.20 Port: 23

### Integration

Message Structure
Command and response messages are standard ASCII text.
? – Request message
! – Control message
# - Error message
~ - Unsolicited message
\n – End of command message, ASCII hex: 0x0A dec: 10

### Protocol

Protocol Command	Description/Response
?Help\n	Lists all possible commands.
?Firmware\n	Request Firmware Version.

	Response: ?Firmware=1.0.0.0\n WebServerSet
?Hostname\n	Request Hostname.  Response: ?Hostname=WattBox\n
?ServiceTag\n	Request the unit's Service Tag.  Response: ?ServiceTag=ST191500681E8422\n
?Model\n	Request Model Number.  Response: ?Model=WB-800-IPVM-6\n
?OutletCount\n	Request Outlet Count.  Response: ?OutletCount=16\n
?OutletStatus\n	Request Outlet States.  Response: ?OutletStatus=0,0,0,0,0,0,0,0,0,0,0,0\n Where the array index plus one is the outlet number and the value at the index indicates state. 0 for off, 1 for on.
?OutletPowerStatus=OUTLET\n  Where OUTLET is the Outlet number.	Request Outlet Power Status for a specific outlet. <i>NOTE: Not supported on WB150/250.</i>  Response: ?OutletPowerStatus=1,1.01,0.02,116.50\n Where 1 is the outlet index you requested, 1.01 is the power in watts, 0.02 is the current in amps, and 116.50 is the voltage in volts.
?PowerStatus\n	Request Power Status for the system. <i>NOTE: Not supported on WB150/250.</i>  Response: ?PowerStatus=60.00,600.00,110.00,1\n Where 60.00 is the current in amps, 600.00 is the power in watts, 110.00 is the voltage in volts, and 1 is the safe voltage status.
!RebootSys\n	Request to reboot the device immediately using Linux system call. The client will lose the connection to the device until the device is back online.   Response: OK\n
?AutoReboot\n	Request Auto Reboot Status for the system.  Response: ?AutoReboot=1\n  Enabled = 1 Disabled = 0

<p>?OutletName\n</p>	<p>Request Outlet Names for all outlets. The names will be sent with brackets around every NAME and comma delimited between each set.</p> <p>Response: ?OutletName={Outlet 1},{Outlet 2},{Outlet 3},{Outlet 4},{Outlet 5},{Outlet 6},{Outlet 7},{Outlet 8},{Outlet 9},{Outlet 10},{Outlet 11},{Outlet 12}\n</p>
<p>!OutletNameSet=OUTLET, NAME\n</p> <p>Where OUTLET is the outlet number and NAME is the new name.</p>	<p>Request to change the name of a specific outlet. Names can have a length up to 31 characters. Spaces are not allowed.</p> <p>Response: OK\n</p>
<p>!OutletNameSetAll={NAME},{NAME},{NAME},{NAME},{NAME},{NAME},{NAME},{NAME},{NAME},{NAME}\n</p> <p>Where NAME is the new name.</p>	<p>Request to change the names for every outlet. Order matters and starts with Outlet 1. The brackets are required around every NAME with commas in-between each set. Names can have a length up to 31 characters.</p> <p>Response: OK\n</p>
<p>?UPSStatus\n</p>	<p>Request UPS Status if there is a UPS attached.</p> <p>Response: ?UPSStatus=50,0,Good,False,25,True,False\n</p> <p>Where 50 is the battery charge percentage, 0 is the battery load as a percentage, Good indicates battery health, False indicates power lost, 25 indicates battery runtime in minutes, True indicates alarm enabled, False indicates alarm muted.</p> <p>Battery Charge: 0-100% Battery Load: 0-100% Battery Health: Good/Bad Power Lost: True/False Battery Runtime: Number in Minutes Alarm Enabled: True/False Alarm Muted: True/False</p>
<p>?UPSConnection\n</p>	<p>Request UPS Connection to find out if a UPS has been attached to the wattbox.</p> <p>Response: ?UPSConnection=0\n</p> <p>Disconnected = 0 Connected = 1</p>
<p>!OutletSet=OUTLET,ACTION,DELAY\n</p> <p>Where OUTLET is the outlet number and ACTION is ON/OFF/TOGGLE/RESET. If action RESET, an optional third</p>	<p>Request to set a specific outlet to a new state. RESET does adhere to power on delay but one may override that value by passing a third optional parameter for DELAY. This delay must be in seconds and ranges from 1 – 600 seconds. To reset all outlets, set OUTLET to 0 and action to RESET. If this command is successful and the outlet is enabled, it will trigger an unsolicited '~OutletStatus' message for each outlet state change.</p>



	<p>Reboot attempts [0 = unlimited, 1-10] - Select a value between 0 and 10. This is the number of times the device will auto-reboot. 0 represents infinite reboots.</p> <p>Response: OK\n</p>
<p>!FirmwareUpdate=URL\n</p> <p>Where URL is the full path to the upgrade file.</p>	<p>Update firmware for the device. This API will respond OK right before the system shuts down. The client will lose the connection to the device until the device is back online.</p> <p>Response: OK\n</p>
!Reboot\n	<p>Request to reboot the device immediately. The client will lose the connection to the device until the device is back online.</p> <p>Response: OK\n</p>
<p>!AccountSet=USER,PASS\n</p> <p>Where USER is the username and PASS is the password.</p>	<p>Request to change the login credentials for a given user and given password. If successful, the client will lose the connection and require a reconnect to login again. Invalid usernames or passwords will be rejected.</p> <p>Response: OK\n</p>
<p>!NetworkSet=HOSTNAME,IP,SUBNET,GATEWAY,DNS1,DNS2\n</p> <p>Where HOST is the hostname, IP is the static address, SUBNET is subnet, GATEWAY is gateway, DNS1 is primary dns server, DNS2 is secondary dns server.</p>	<p>Request to change the network settings for the device.</p> <p>HOSTNAME only allows characters A-z,0-9, and '-'</p> <p>If setting DHCP, do not send IP,SUBNET,GATEWAY,DNS1,DNS2.</p> <p>If setting STATIC, IP,SUBNET,GATEWAY,DNS1 are required. DNS2 is optional and will be auto filled to 8.8.8.8 if nothing was entered.</p> <p>If the settings are valid, the changes will be made and the device will reboot. The client will lose the connection to the device until the device is back online. Please note the device may come back at a different IP address depending on the settings sent.</p> <p>Response: OK\n</p>
<p>?NetworkGet\n</p> <p>Minimum FW v2.3.0.8</p>	<p>Request the network settings for the device</p> <p>Response: ?NetworkGet= &lt;DHCP&gt;,&lt;STATIC_DNS&gt;,&lt;HOSTNAME&gt;,&lt;IP&gt;,&lt;SUBNET&gt;,&lt;GATEWAY&gt;,&lt;DNS1&gt;,&lt;DNS2&gt;\n</p> <p>DHCP: 0 or 1, 1 – enabled, 0 - disabled STATIC_DNS: 0 or 1, 1 – enabled, 0 - disabled</p>

	<p>HOSTNAME: Up to 64 characters</p> <p>IP: x.x.x.x</p> <p>SUBNET: x.x.x.x</p> <p>GATEWAY: x.x.x.x</p> <p>DNS1: x.x.x.x</p> <p>DNS2: x.x.x.x</p>
<p>!ScheduleAdd={NAME},{OUTLET, OUTLET,OUTLET},{ACTION},{FREQ},{DAY,DAY,DAY   DATE},{TIME}\n</p> <p>Where NAME is the schedule name, OUTLET is an array of outlet numbers, ACTION is the action performed, FREQUENCY is once or recurring, DAYS or DATE are the days for recurring or date for once, TIME is the time.</p>	<p>Request to add a schedule to the scheduled events for the device. Brackets are required around every value with a comma delimiter between each set. Parameter 2 is an array of outlets to indicate which outlets the ACTION will be performed on once the schedule is triggered. Reference the below table to determine the correct field and send the corresponding values. Values outside of any of these ranges will be rejected.</p> <p>Outlet = {1,2,3} Would tie outlets 1,2,3 to this schedule.</p> <p>Action</p> <ul style="list-style-type: none"> <li>- Off = 0</li> <li>- On = 1</li> <li>- Reset = 2</li> </ul> <p>Frequency</p> <ul style="list-style-type: none"> <li>- Once = 0</li> <li>- Recurring = 1</li> </ul> <p>If Recurring</p> <p>Days [s,m,t,w,t,f,s] – This is an array where the index indicates the day of the week and if the value at the index is a 0, the day is not included, if the value at the index is 1, the day is included. The following example will recur every Monday, Wednesday, and Friday. {0,1,0,1,0,1,0}</p> <p>If Once</p> <p>Date [yyyy/mm/dd] – {2018/09/28}</p> <p>Time [hh:mm] – 24-Hour based so 1:30pm would be represented as 13:30.</p> <p>Response: OK\n</p>
<p>!HostAdd=NAME,IP,{OUTLET,OUTLET}\n</p> <p>Where NAME is the host name, IP is the Website or IP address to be tested, and OUTLET is an array of outlet numbers.</p>	<p>Request to add a host to the list of hosts to be monitored by the device. Brackets are required around the outlets array. This array indicates which outlets should be tied to the host being added.</p>
<p>!SetTelnet=MODE\n</p> <p>Where mode is 0 for disabled, 1 for enabled.</p>	<p>Request to enable or disable the telnet service. A reboot is required for settings to take effect. All MODE values other than 0 or 1 will generate error message.</p> <p>Response:</p>

	OK\n
!WebServerSet=MODE\n  Where mode is 0 for disabled, 1 for enabled	Request to enable or disable the web server. A reboot is required for settings to take effect. All MODE values other than 0 or 1 will generate error message.  Response: OK\n  Requires WattBox firmware 2.0
!SetSDDP=MODE\n  Where mode is 0 for disabled, 1 for enabled	Request to enable or disable SDDP broadcasting. All MODE values other than 0 or 1 will generate error message.  Response: OK\n  Requires WattBox firmware 2.0
!Exit	Close the session gracefully.
~OutletStatus=STATE, STATE, STATE, STATE, STATE, STATE, STATE, STATE  Where STATE can be 0 for OFF or 1 for ON.	Unsolicited outlet status message. This message is generated anytime an outlet changes its state. Where the array index plus one is the outlet number and the value at the index indicates STATE.
#Error\n	Sent whenever an invalid command was received, or an internal device error has occurred. Please see the device log page for further detailed error messages.
?FaceplatePresent  <i>Minimum FW v2.3.0.2</i>	Responds 0 or 1 to indicate presence
?FaceplatePort  <i>Minimum FW v2.3.0.2</i>	Responds 0 if no faceplate, 1, 2 or 3 to indicate a faceplate port.
?FaceplateLedLevel  <i>Minimum FW v2.3.0.2</i>	Responds with a level between 0 and 65535. 0 is dimmest.
!FaceplateLedLevelSet=LEVEL  <i>Minimum FW v2.3.0.2</i>	Level should be an integer between 0 and 65535. 0 is dimmest.
?FaceplateUUID  <i>Minimum FW v2.3.0.2</i>	Responds with the UUID string
?UPSVoltageRange  <i>Minimum FW v2.3.0.6</i>	Responds with N, W, G or U N – Normal

	<p>W – Wide</p> <p>G – Generator</p> <p>U – Unknown (ask again, UPS doesn't always respond)</p>
<p>!UPSVoltageRange=RANGE</p> <p><i>Minimum FW v2.3.0.6</i></p>	<p>Set RANGE to one of:</p> <p>N – Normal</p> <p>W – Wide</p> <p>G – Generator</p>
<p>?AdapterSensorData</p> <p><i>Minimum FW v2.6.2.0</i></p>	<p>Fetches Sensor Data for all WB-ACC-ADAPTER-800's. Each entry is separated by a semicolon. If no adapters are connected, then 'None' is returned.</p> <p>Example Response:</p> <p>?AdapterSensorData= &lt;a1_tag&gt;,&lt;a1_port&gt;,&lt;s1_conn&gt;,&lt;s1_unit&gt;,&lt;s1_data&gt;,&lt;s2_conn&gt;,&lt;s2_unit&gt;,&lt;s2_data&gt;</p> <p>&lt;a*_tag&gt; - Service Tag of the connected WB-ACC-ADAPTER-800</p> <p>&lt;a*_port&gt; - Port the device is connected to. 0 – Directly connected</p> <p>&lt;s*_conn&gt; - 'True' means sensor probe is connected, 'False' otherwise.</p> <p>&lt;s*_unit&gt; - 0 = Sensor units are tenths of degrees Celsius.</p> <p>&lt;s*_data&gt; - Sensor data value in &lt;s*_units&gt;. Data is invalid if &lt;s*_conn&gt; is 'False'.</p>
<p>?AdapterServiceTags</p>	<p>Fetches all connected WB-ACC-ADAPTER-800 service tags. If no adapters are connected, then 'None' is returned.</p> <p>Example Response:</p> <p>?AdapterServiceTags=STXXXXXXXXXXXXXXXXXX</p>
<p>?AdapterConfig=&lt;service_tag&gt;</p>	<p>Fetches the config for a specific WB-ACC-ADAPTER-800.</p> <p>Example Response:</p> <p>?AdapterConfig=&lt;button_type&gt;,&lt;button_polarity_when_latching&gt;,&lt;dc_out_pin_mode&gt;</p> <p>&lt;button_type&gt; configures what kind of button or signal is connected to the 'Trigger' Pin on the phoenix connector. 0-Momentary Button, 1-Latching Button</p>

	<p>!&lt;button_polarity_when_latching&gt;</p> <p>&lt;button_polarity_when_latching&gt; determines which part of the signal is active. 0-ActiveLow, 1-ActiveHigh. Only applies when &lt;button_type&gt; is latching.</p> <p>&lt;dc_out_pin_mode&gt; determines the behavior of the 'DC OUT' pin on the phoenix connector. 0-Disabled, 1-Enabled, 2-ToggleDuringAction</p>
!AdapterConfig=<service_tag>,<button_type>,<button_polarity_when_latching>,<dc_out_pin_mode>	<p>Sets the config for a specific WB-ACC-ADAPTER-800.</p> <p>Example Response: OK</p>

Example:

```
$ nc 192.168.26.27 23
Please Login to Continue
Username: wattbox
Password: wattbox
Successfully Logged In!
?Firmware
?Firmware=0.0.0.1
|
```