

SNAP Manual v1.04t



Overview

The screenshot displays the SNAP configuration interface with several panels. The top left panel, titled "Inovonics Input", shows configuration for an input method with fields for Name, Default Destination, Maintenance Destination, Report Destination, Send Report At, No Check In Delay, and Use Custom Tamper Messages. Below this is a "State" panel with fields for Repeat After, Repeat Suffix, Escalate After, Escalate Destination, Escalation Suffix, and Process. The top right panel, titled "Serial Input", shows configuration for a serial input method with fields for Port, Settings, and Flow Control. The bottom left panel, titled "TAP Output", shows configuration for a tap output method with fields for Name, Cancel Word, and Modifiers. The bottom right panel, titled "Ethernet Output", shows configuration for an ethernet output method with fields for IP Address and IP Port.

SNAP System

The SNAP system is designed to perform the following types of functions:

- Alert detection using [input protocol](#) processing
- Alert generation using browser based free-form message entry with multi-selection of alert recipients and optional selection of predefined messages
- Alert generation using alert monitoring rules applied to incoming alerts, with or without assigned [schedules](#), defined in [Monitor Modifiers](#)

- Alert generation by identifying maintenance class alerts where possible, such as tamper events, low sensor battery conditions, etc.
- Alert event processing, including:
 - Alert repeat notification and/or escalation, using alert state rules, defined in [State Modifiers](#)
 - Translate either alert messages or alert destinations using translation rules defined in [Translate Modifiers](#)
 - Route alerts to alternate or multiple destinations using routing rules defined in [Route Modifiers](#)
 - Filter out alerts, by alert message or destination, using either white-list or black-list methods, with or without assigned [schedules](#), defined in [Filter Modifiers](#)
 - Smooth out alert notification, using [Debounce Modifiers](#)
 - Alert cancellation detection, using cancel keyword detection, and group cancellation methods using [Group Cancel Modifiers](#)
- Alert notification using one or more output protocols, using serial port, Ethernet, and/or audio communications
- Alert notification using digital television
- Logging of system activity
- Protocol conversion
- Data filtering
- Data translation
- Data output splitting, e.g. one-to-many data processing method
- Data input combining, e.g. many-to-one data processing method
- E-mailed report generation using [Report Modifiers](#)
- Message encryption, using [Encryption Modifiers](#)

SNAP User License and Limitations

When SNAP is used for healthcare applications, by installing and configuring SNAP, the user agrees that SNAP has no UL certifications and is designed to operate only as a secondary alerting system for senior living and/or assisted living applications, and that SNAP should not be installed at sites where the user is not already operating a primary alerting system.

SNAP offers a browser based configuration interface which uses the [Google Blockly library](#) to allow a rich configuration interface in one workspace.

When connecting to SNAP via browser, you may be prompted for a userid and password. You can use the following userid values to login:

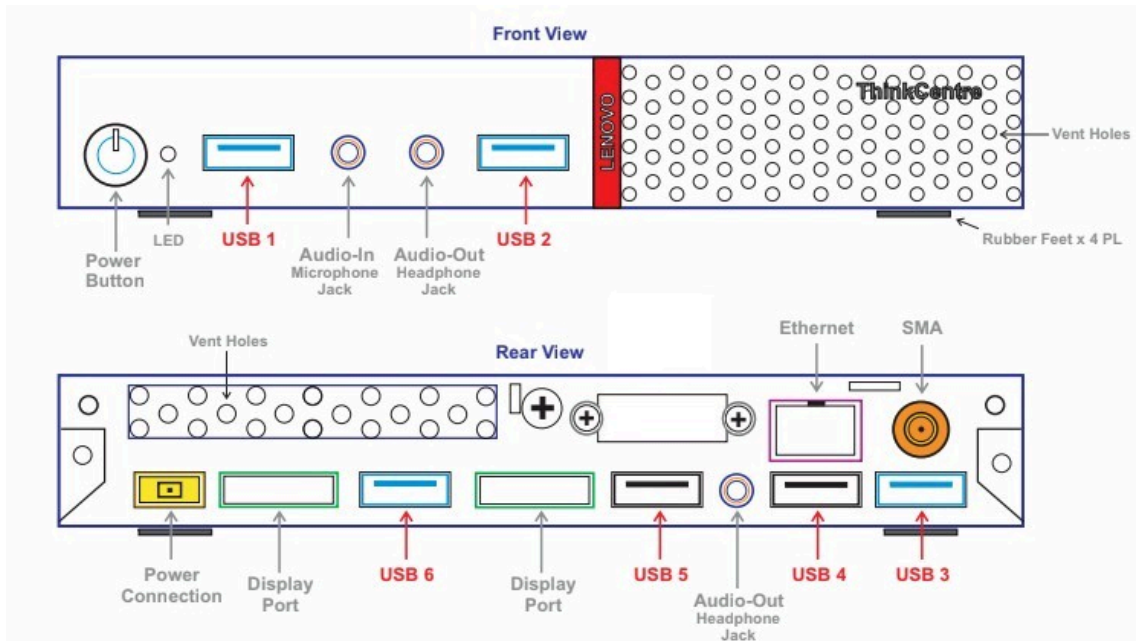
- **admin** - Logs you into the SNAP Configuration Workspace. The default admin password value is password, and is editable in the [SNAP System Configuration](#) page.
- **messaging** - Logs you into the SNAP Messaging Client. The default messaging password value is password, and is editable in the [Messaging Client Input](#) block.
- **alerts** - Logs you into the Alerts List display. The default alerts password value is password, and is editable in the [Alerts List Output](#) block.
- **recover** - Logs you into the SNAP Configuration Workspace. The recover password value never changes and is printed on a label on your SNAP unit, as well as being displayed in the [SNAP System Configuration](#) page.

- **user** - Allows login to certain SNAP configuration tables. The default user password value is password, and is editable in the [SNAP System Configuration](#) page.

User logins can be setup as [desktop shortcuts](#), where the user double clicks to launch a connection to a specific configuration table.

SNAP is designed to be offered on a Lenovo Thin Client computer

Following is a USB input layout for the SNAP Lenovo Thin Client unit:



The SNAP system can be purchased to offer specific operating modes to fit various markets:

- **SNAP Standard Mode** - The full featured system that is described in this document
- **SNAP Lite Mode** - A limited feature mode to fit smaller budgets. The Lite mode is upgradeable to the Standard mode. The Lite mode includes the following initial features:
 - All protocols offered in the Standard model
 - 3 I/O port license
 - 1 Pager App connection license
- **SNAP STG Mode** - A limited feature mode that is compatible with the Rauland Responder 5 and 5000 (tm) nurse call systems. The STG mode is upgradeable to the Standard mode. The STG mode includes the following initial features:
 - Rauland SIP Input protocol
 - TAP Output protocol
 - 5 I/O port license
 - 1 Pager App connection license

SNAP data inputs may use multiple [input methods](#), and include the following standard protocols:

- [Plain Text](#)
- [COMP2](#)
- [TAP](#)

- [Inovonics](#)
- [HTTP](#)
- [Rauland SIP](#)

SNAP data outputs may use multiple [output methods](#), and include the following standard protocols:

- [COMP1](#)
- [COMP2](#)
- [E-mail](#)
- [TAP](#)
- [JSON](#)
- [Pager App](#)
- [Alerts List Output](#)
- [HTTP Output](#)

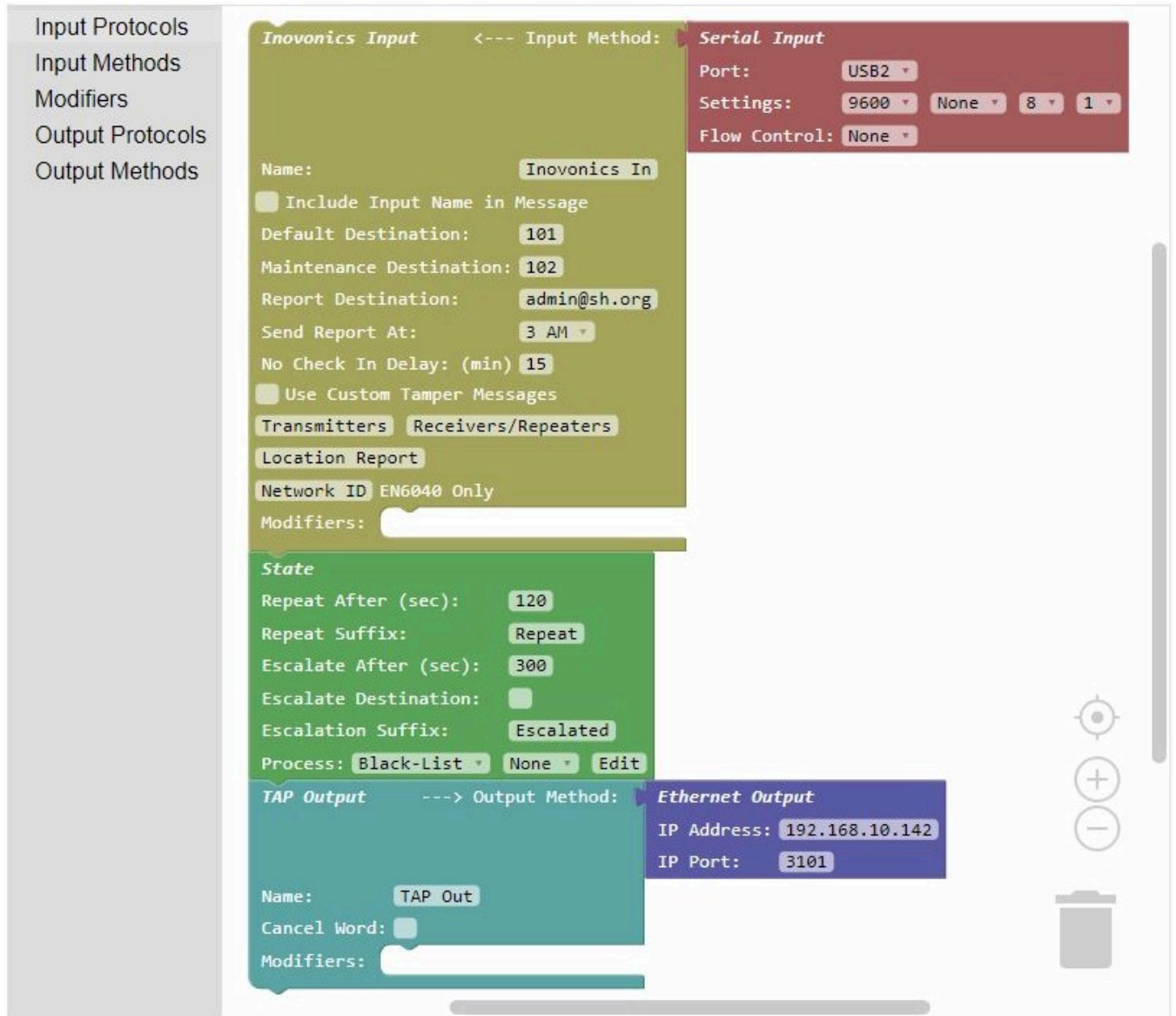
Note that when using the Pager App Output with the Pager App, additional licensing may be required to allow those App connections to occur. Note that Pager App licensing is independent and the licenses cannot be inter-mixed between apps.

Note that when using the Inovonics Input, additional licensing may be required to allow processing of those additional Inovonics transmitters to occur.

Additional protocols and features that may be licensed for use in the SNAP product include the following:

- Additional Input/Output capacity (Available only for SNAP Standard model)
- Additional concurrent Pager App Connections
- Additional Inovonics Transmitters
- [CCTV Text Overlay Output](#)
- [Accutech Residentguard Input](#)
- [Audio Output](#)
- [E-mail Input](#) (Available only for SNAP Standard model)
- [Messaging Client Input](#) (Available only for SNAP Standard model)
- [Alerts List Output Zoning](#) (Available only for SNAP Standard model)
- [Rauland SIP Input](#) (Available only for SNAP Standard model)
- [Rauland RSI Output](#) (Available only for SNAP Standard model)
- [Adaptive Displays Output](#) (Available only for SNAP Standard model)

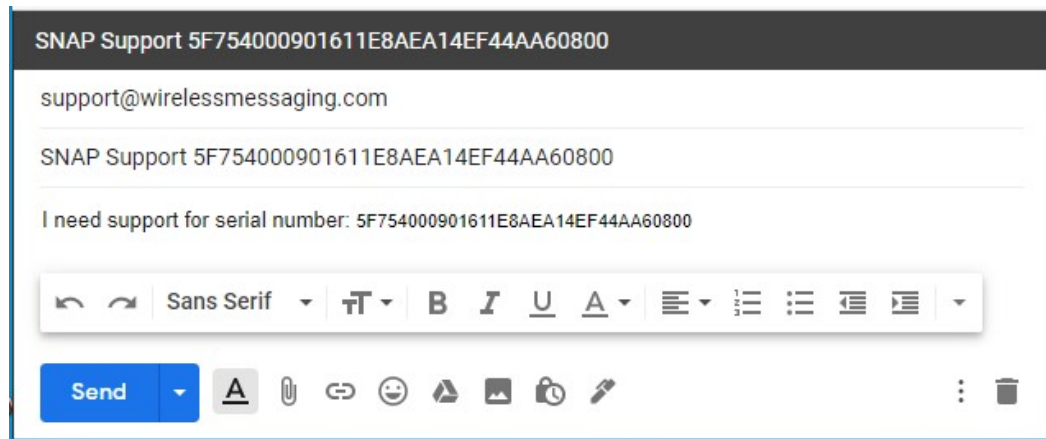
SNAP Configuration Workspace



SNAP configuration is composed of a sequence of blocks that are snapped together within the SNAP configuration workspace, with each block configured to the requirements of the system, and as constrained by the SNAP license registration system.

The SNAP configuration workspace is composed of the SNAP Toolbox on the left, the zoomable block structure workspace in the center, the button menu on the top left, the status and logout area to the top right, and the system serial number at the bottom.

To launch an e-mail to the support department, you can click on the system serial number at the bottom of the configuration workspace. Edit the body of the support e-mail to tell us what you would like to do.



To start creating a SNAP system configuration, click on the Toolbox menu items to fly out a list of blocks for each menu item, then click and drag an object to the workspace. Input Protocols, Modifiers, and Output protocols snap into each other on their top and bottom surfaces. Input Methods and Output Methods snap into the right side of Input Protocols and Output Protocols, where applicable. Modifiers can also be snapped into the Modifiers sections of the Protocol blocks, to support modification of only the alerts inside the associated Protocol block.

The SNAP workspace bottom right provides a target icon for centering your SNAP configuration structure. It also contains plus and minus icons for adjusting the zoom level in the workspace. Note that zooming also affects the size of the blocks appearing in the flyout menu. Scrolling your mouse scroll wheel while hovering over the SNAP workspace zooms the block structure in and out.

To remove blocks from your SNAP system configuration, first separate and isolate any blocks that you want to delete. The SNAP workspace contains a trash can, so that you can drag one or more blocks to the trash can for deletion. Another method of deleting blocks is to drag one or more blocks into the gray flyout menu area to the left. To delete fewer blocks than the entire structure, separate the connected blocks by clicking and dragging one or more blocks away from the block structure. All blocks in any connected group of blocks being dragged will all be deleted together.

You can discover the following useful features if you right click any block on the SNAP workspace:

- **Duplicate** - Duplicate the selected block
- **Add Comment** - Make a comment on this block, which creates a clickable icon on the block, so that the comment is stored with your block structure
- **Delete Block** - Delete the selected block
- **Help** - Launch specific help for the selected block

Your SNAP system configuration tries to be processed by the SNAP server in real time. If you click the Check to Pause Unit checkbox in the top right, you can pause SNAP server operation while doing heavy configuration operations.

SNAP Configuration

Configuration of SNAP system consists of defining the following:

- One or more [input protocols](#) with associated [input method](#)
- Zero or more [modifiers](#)

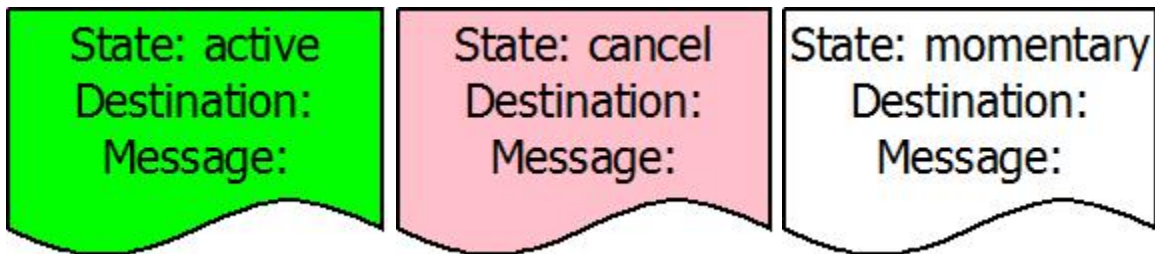
- One or more [output protocols](#) with associated [output method](#)

SNAP converts incoming data into internal alert objects that are composed of the following elements:

- **Message** - The message value that is delivered to alert recipients
- **Destination** - The value that defines how to route alerts to the appropriate recipients. Destination values can be numeric, representing pager ID, PIN, address, etc., or destination values can be alphanumeric, such as e-mail addresses or easy to remember reference values, such as **east wing**.
- **State** - State values can represent states of **active**, **cancel**, or **momentary**, which affect downstream alert object processing. An alert with a momentary state has no tracked alert time duration and never has a cancel state associated with it, but it is still reported. Alerts with state values of **active** or **canceled** are considered **stateful** alerts, while those with a state value of **momentary** are considered **stateless**.

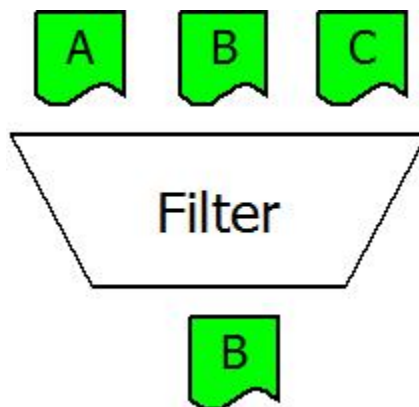
Stateful alerts allow:

- Persistent notification of active alerts
- Auto-removal of alerts from active alert lists upon cancellation state
- Reporting of alert event timestamps and durations
- Stateless alerts allow:
 - One-time notification of alert events
 - Reporting of alert event timestamps

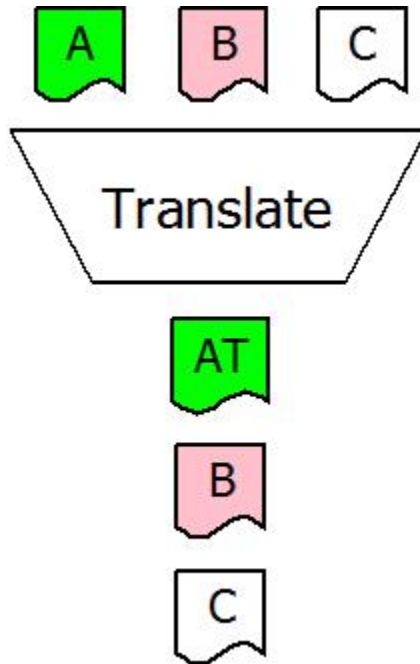


The SNAP internal alert objects are modified and filtered as required by SNAP [modifiers](#), then the remaining alert objects are converted to the applied [output protocols](#) and associated [output methods](#).

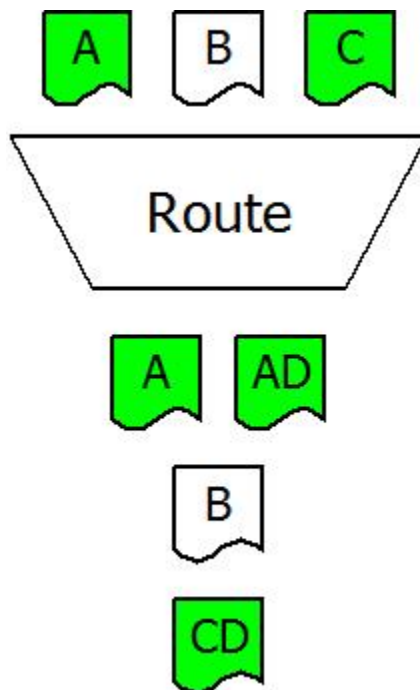
The Filter Modifier diagram below, filters out some alert objects



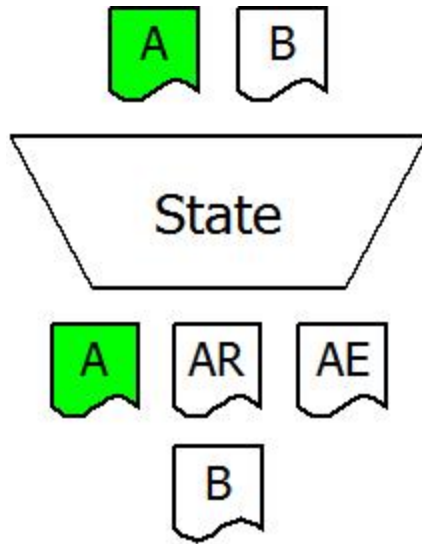
The Translate Modifier diagram below, translating some alert object message values while passing through all other alert objects



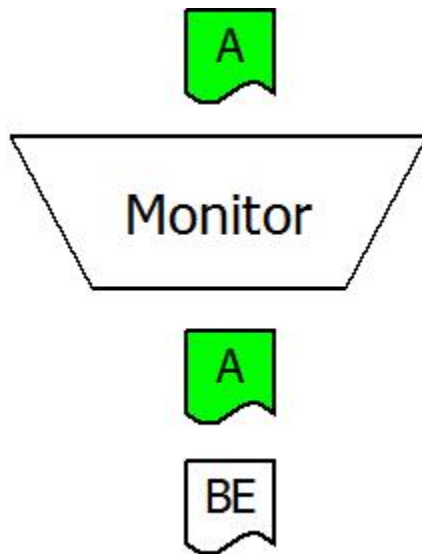
The Route Modifier diagram below, changing some alert object destination values, creating some cloned alerts with new destination values, while passing through all other alert objects



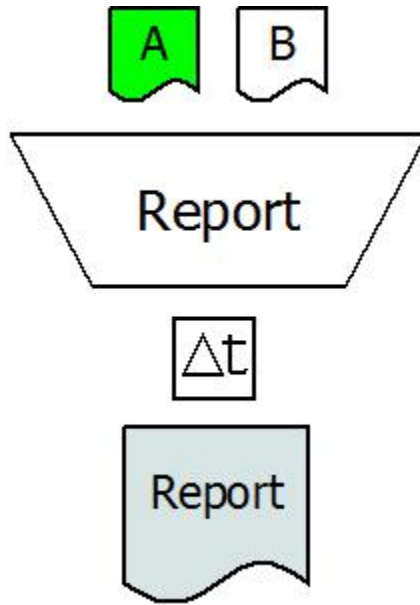
The State Modifier diagram below, creating new repeat and escalation alert objects while passing through all alert objects



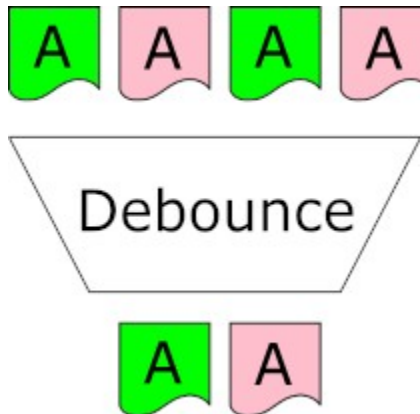
The Monitor Modifier diagram below, creating new exception alert objects while passing through some or all alert objects



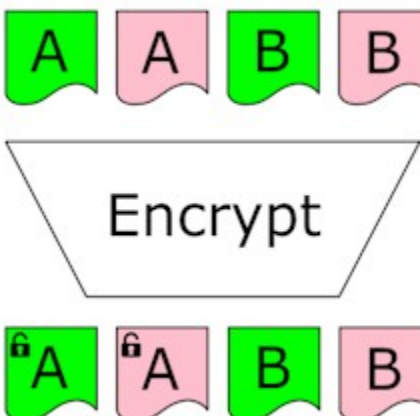
The Report Modifier diagram below, creating daily, scheduled, e-mailed reports as a special modifier, and is dependent on use of the [E-mail Output Block](#)



The Debounce Modifier diagram below, smoothing out the generation of activation and cancellation events for particular alerts



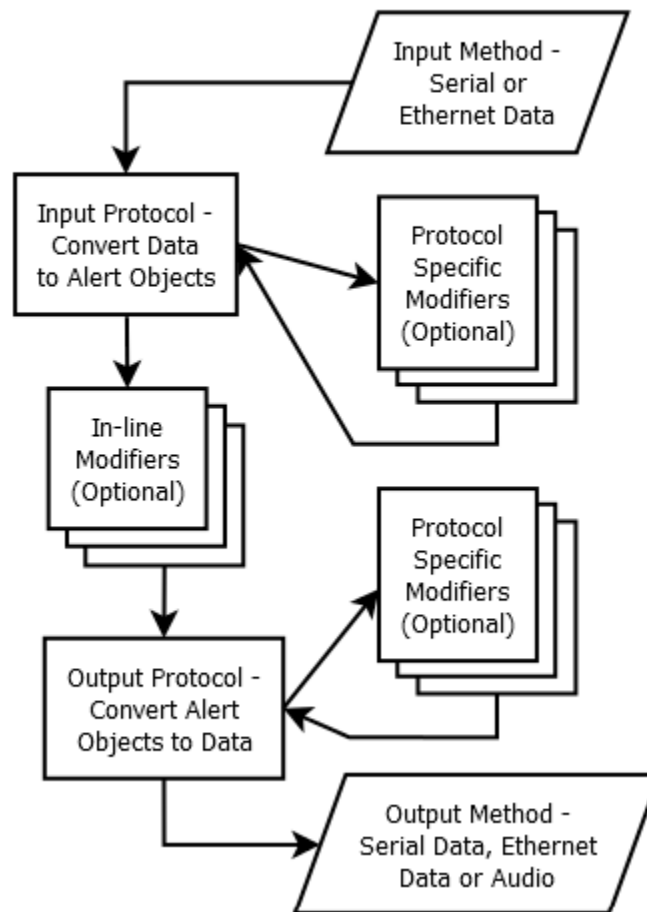
The Encryption Modifier diagram below, tagging alert message values for encryption, to occur in particular alerts, for encryption in certain output blocks



The following SNAP Processing Model is applied in SNAP between any one specific Input Protocol and specific Output Protocol in the system. If a SNAP system is configured with more than one Input Protocol or more than one Output Protocol, multiple instances of the SNAP Processing Model are applied in parallel. The SNAP system creates a SNAP Processing Model for each combination of Input Protocol and Output Protocol relationship defined in the system. For example, as shown in the SNAP configuration image further below, if the SNAP system is configured with both a TAP Protocol Input and an Inovonics Protocol Input, in addition to a TAP Protocol Output, then the SNAP system will create a SNAP Processing Model for the TAP In to TAP Out relationship, and create another SNAP Processing Model for the Inovonics In to TAP Out relationship.

If the SNAP configuration includes one or more In-line Modifier blocks, those In-line blocks will be applied to all Input Protocol blocks that are above that In-line Modifier.

Processing Model



Each block feeds data into the next block in a top to bottom sequence. Therefore, your Input Protocol Blocks should be stacked at the top of the configuration and your Output Protocol Blocks should be stacked at the bottom. Modifier blocks can be applied either between any blocks or stacked in the Modifiers portion of any Input or Output block.

When Modifier blocks are stacked inside the Modifiers portion of an Input or Output block, those Modifier blocks only affect SNAP alert objects that are generated inside the affected Input or Output block, and will not affect any alert objects passed to the affected Input or Output block from blocks above that block. When

Modifier blocks are placed between blocks, the Modifier blocks affect all alert objects passing through the system from all blocks above that Modifier block.

In general, all alert objects that enter a block are also output from the block. Some blocks, in addition to all [Input Protocol blocks](#), can generate new alert objects, including:

- [Route Modifier block](#)
- [State Modifier block](#)
- [Monitor Modifier block](#)
- [Report Modifier block](#), which creates a specialized alert object whose destination is an e-mail address and whose message value is an HTML formatted report
- [Group Cancel Modifier block](#), which creates specialized alert objects with a cancel state, whose purpose is to allow downstream systems that maintain alert state to be able to automatically remove individual alerts from their active alerts list

Some blocks can remove alert objects as they pass through the block, including:

- [Filter Modifier block](#)
- [Monitor Modifier block](#)
- [Debounce Modifier block](#)

The following image shows a SNAP configuration with both TAP and Inovonics inputs, with a State Modifier block inside an Inovonics Protocol Input block, which demonstrates the State Modifier affecting only the alerts generated inside the Inovonics Protocol Input block. Note that the TAP Protocol Input generated alert objects are not affected by the State Modifier block in this example, because as shown in the SNAP Processing Model above, a given relationship between a specific Input Protocol and a specific Output Protocol are only affected by In-Line Modifiers or Modifiers applied specifically to the affected Protocol block.

TAP Input <--- Input Method: **Ethernet Input**
IP Port: 4322

Name: TAP In
Cancel Word:
 Include Input Name in Message
Modifiers:

Inovonics Input <--- Input Method: **Serial Input**
Port: USB2
Settings: 9600 None 8 1
Flow Control: None

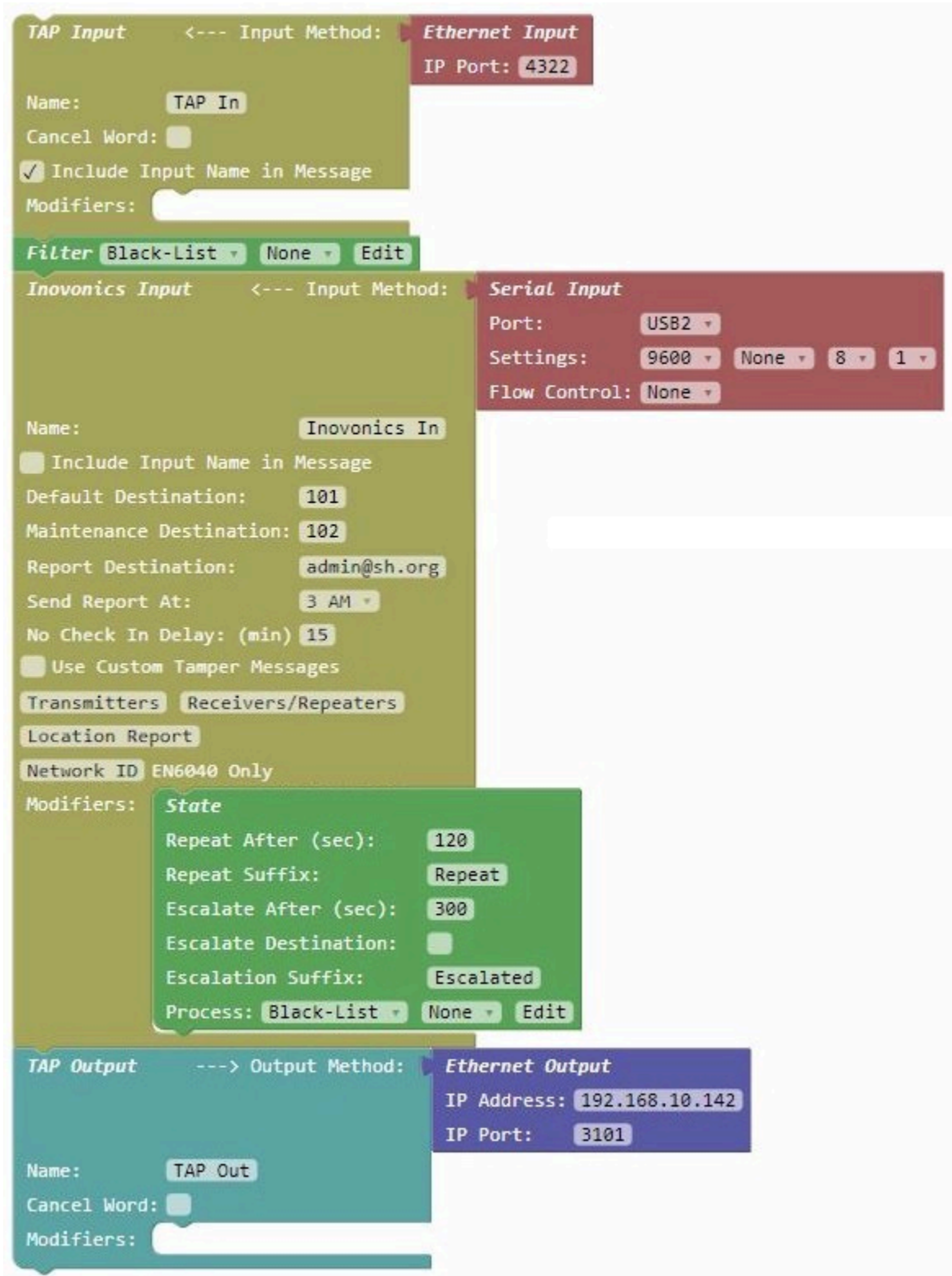
Name: Inovonics In
 Include Input Name in Message
Default Destination: 101
Maintenance Destination: 102
Report Destination: admin@sh.org
Send Report At: 3 AM
No Check In Delay: (min) 15
 Use Custom Tamper Messages
Transmitters Receivers/Repeaters
Location Report
Network ID EN6040 Only

Modifiers: **State**
Repeat After (sec): 120
Repeat Suffix: Repeat
Escalate After (sec): 300
Escalate Destination:
Escalation Suffix: Escalated
Process: Black-List None Edit

TAP Output ---> Output Method: **Ethernet Output**
IP Address: 192.168.10.142
IP Port: 3101

Name: TAP Out
Cancel Word:
Modifiers:

The following image shows application of an In-line Filter Modifier that only affects the alert objects created by the TAP Protocol Input block



The following image shows application of stacked In-line Filter and State Modifiers that affect the alert objects created by both the TAP Protocol Input block and the Inovonics Protocol Input block.



Testing SNAP

You can test and observe SNAP configurations using several methods, including the following:

- [Input Simulator](#)
- [Capture Output Method](#)

- [System Logs](#)

Note: In some cases, when reconfiguring USB serial port assignments and related device connections, when you notice unexpected USB serial port related behavior, the SNAP system may need to be restarted. You can restart the SNAP system using either the **Reboot System** link in the [SNAP System Configuration](#) page, or using the system power button.

System

System Settings

[Close](#)

System Configuration Page

HELP

Version: 1.02
System IP: 192.168.10.115

DHCP Enabled

Static IP: (Only used if DHCP is NOT ENABLED)
 Subnet Mask: (Only used if DHCP is NOT ENABLED)
 System Gateway: (Only used if DHCP is NOT ENABLED)

Name Server:
 Time Server:

Time Zone:

Host Name:

Admin Password:
 User Password:

Reg Key:

Site Name:

[Export System Database](#) [Import System Database](#)

[Update System Software via Internet](#) OR [Update System Software via Zip File](#)

[Upload Custom Alert Sound \(.wav or .mp3\)](#)

[Reboot System](#)

Authorizations

Feature	Authorizations
Max Total Inputs/Outputs	5
Max Inovonics Transmitters	50
Max Concurrent Pager App Connections	3
Messaging Client Input	Enabled
E-Mail Input	Enabled
Accutech Input	Enabled
CCTV-POS Output	Enabled
Alerts List Zones	Enabled
Audio Output	Enabled

Recovery Credentials
 User Name: **recover**
 Password: **73F3K2PF**

Click on the **Settings** button to display the System Configuration Page. Note the **Close** link in the top left of the popup window, as well as the System IP value at the upper left, indicating the currently assigned IP address.

SNAP ships with DHCP enabled. Note that the DHCP Enabled checkbox must be unchecked if you want to assign specific Static IP, Subnet Mask, and System Gateway values.

The **Name Server** field is useful when it is important which name server that SNAP should use to help resolve domain based addressing. Note that if the DHCP Enabled checkbox is unchecked, that if the Name Server field is blank, the system will apply a default name server value.

The **Time Server** field is useful when system log timestamping is important. Note that if the DHCP Enabled checkbox is unchecked, that if the Time Server field is blank, the system will apply default time server values.

The **Time Zone** field should be set to the time zone that SNAP resides in.

The **Host Name** field is useful when discovering SNAP on your network using the Discover and Reset tool.

The **Admin Password** field allows configuration of the password used with the **admin** userid value when logging into the SNAP configuration workspace.

The **User Password** field allows configuration of the password used with the **user** userid value when logging into SNAP configuration tables.

The **Reg Key** field is where you paste any new SNAP Registration Key value if you upgrade your system. The Request Key field is the value that you use during the process of requesting a SNAP upgrade.

The **Site Name** field is used with certain data output structures to help identify which SNAP server generated the data

The **Save** button saves any new system configuration values.

The **Update Registration Key** button allows you to update the SNAP Reg Key value if you are in the process of upgrading your system. The Upgrade Registration Key process requires SNAP to have internet access. If Internet access is unavailable, you can paste a new registration key value into the Reg Key field, then click the Save button.

The **Export System Database** link allows you to export your full current SNAP workspace configuration to a db file extension. The **Import System Database** link allows you to import a SNAP db file into your SNAP system.

It is **highly recommended** that you export your SNAP system database to a safe location once you are satisfied with the configuration, as a means of backing up your work. If your current system database becomes corrupted, you can import your backed up database to restore operation.

Note that the Import System Database process will attempt to replace the SNAP configuration structure currently in your SNAP workspace. It is useful to be able to send SNAP db files between SNAP users.

Note that the SNAP workspace will be paused after an Import System Database process occurs, which will require you to check the Unit Paused checkbox to unpauses and restore system operation.

The **Update System Software via Internet** link allows the SNAP system to try to update its software, if there is an internet connection available for the unit

The **Update System Software via Internet** link allows the SNAP system to try to update its software, if there is an internet connection available for the unit

The **Upload Custom Alert Sound (.wav or .mp3)** link allows you to upload a **wav** or **mp3** file from your network to the SNAP system, to allow you to output a custom alert sound, which replaces the default **chime** sound.

The **Reboot System** link allows you to reboot the SNAP system, as required, to apply the key network settings that only get applied at startup


The **Authorizations** table identifies the authorized functions allowed in SNAP by the installed Registration Key

The **Recovery Credentials** section provides unique recovery login credentials for each SNAP system, for use if the admin user password information becomes lost. Please document the recovery credentials somewhere safe.

System Schedules

Click on the **Schedules** button to display the Schedules Page. SNAP schedules can be optionally applied in the following areas of the SNAP system:

- **Filter Modifier** - To selectively apply a schedule to the SNAP alert object filtering process
- **Monitor Modifier** - To selectively apply a schedule to the SNAP alert object Monitor rules processing

Schedules		
Filter :type any text here	+ Add New Schedule HELP	
Name	Edit	Delete
schedule	Click Here to Open	

To create a new schedule, click the Add New Schedule button, enter a Schedule Name value, then click OK to create a new Schedule line item. Once the new Schedule line item is created, you can edit the Name field, as required, and hit the Enter key to save the value.

To edit a schedule item, click on the Click Here to Open link in the Edit column of the Schedules table, which will cause the editing tool for your selected schedule to open.

Schedule: schedule 1



An Alert Processing Schedule is defined by adjusting time sliders and assigning one or more days of the week to a particular schedule row. Each schedule row can be configured to have up to three separate processing ranges within each 24 hour period defined in a day. Each schedule is defined for one or more of the days of the week, where the schedule is applied based on what the current time and day is. To have a different schedule for a different day, click on the green plus button to open a new schedule row. You can create up to three schedule rows per schedule. To reduce the quantity of schedule rows assigned to the schedule, click the red minus button.

Assigned days are the day assignment buttons with a white background. Once you assign a day for a particular schedule row, you cannot assign that same day to another schedule row until you unassign that day from a schedule row. You can create a schedule with one or more days unassigned. Alert processing will not occur on unassigned days for a particular schedule.

A processing range is a span of time defined within a schedule, indicated by the blue areas with handles on each end. Alerts and/or recipients associated with the schedule, are processed by the server when an alert occurs at a time that falls within a defined processing range. Each daily schedule is composed of three processing ranges, which can either be joined or separated. When processing ranges are joined and separated, you can see the change in the defined time span text above the schedule slider. To adjust a time range, you can click and drag a range slider handle or click anywhere along the slider. Clicking along the slider will cause a handle for that slider to move to where you clicked.

When a particular handle on a slider is selected by clicking the handle, the handle has a blue highlight around it. Clicking the slider away from a handle causes slider adjustment, based on which handle is currently selected. You can observe the text above the slider to see what effect your selected slider has. To eliminate a gap in the schedule, adjust your handles to overlap each other.

Schedule edits are automatically saved as you make adjustments to the schedule.

System Logs

Click on the **Logs** button to display the System Log Page. The Log page displays approximately 1,000 transaction records, for purposes of troubleshooting.

Following is a sample from the System Log page:

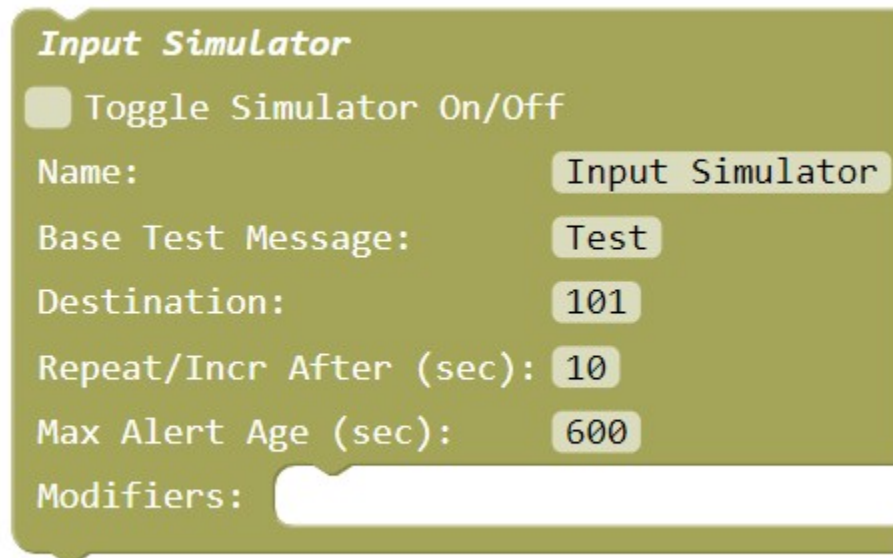
```
2018-05-14 22:39:49> =====Starting SNAP Interpreter=====
2018-05-14 22:39:49> Updating System Configuration.
2018-05-14 22:39:49> Maximum Inputs and Outputs limited to 3.
2018-05-14 22:49:24> Updating System Configuration.
2018-05-14 22:49:24> Maximum Inputs and Outputs limited to 3.
2018-05-14 22:49:24> Examining Path #1.
2018-05-14 22:49:24> Path 1 Layer 1 Contains Block of Type 'input_protocol_simulator'.
2018-05-14 22:49:24> Path 1 Layer 2 Contains Block of Type 'output_protocol_comp2'.
2018-05-14 22:49:24> Path 1 Layer 2 Contains Block of Type 'test_output'.
2018-05-14 22:49:24> Creating New Object: TYPE='input_protocol_simulator' ID='QYcY9GR@Cu{Z.JLj?mmX'
2018-05-14 22:49:24> Creating New Object: TYPE='output_protocol_comp2' ID='c5MH+mKK`DaprK#lj;7P'
2018-05-14 22:49:28> Updating System Configuration.
2018-05-14 22:49:28> Maximum Inputs and Outputs limited to 3.
2018-05-14 22:49:28> Examining Path #1.
2018-05-14 22:49:28> Path 1 Layer 1 Contains Block of Type 'input_protocol_simulator'.
2018-05-14 22:49:28> Path 1 Layer 2 Contains Block of Type 'output_protocol_comp2'.
2018-05-14 22:49:28> Path 1 Layer 2 Contains Block of Type 'test_output'.
2018-05-14 22:49:29> IN: Input Protocol Simulator 'Input Simulator': [101]-"Test message 001"-MOMENTARY
2018-05-14 22:49:29> OUT: 'COMP2 Out' Encoded [101]-"Test message 001"-MOMENTARY
2018-05-14 22:49:29> OUT: Saving to Test Output File: 101<CR><LF>Test message 001<CR><LF>
2018-05-14 22:49:44> IN: Input Protocol Simulator 'Input Simulator': [101]-"Test message 002"-MOMENTARY
2018-05-14 22:49:44> OUT: 'COMP2 Out' Encoded [101]-"Test message 002"-MOMENTARY
2018-05-14 22:49:44> OUT: Saving to Test Output File: 101<CR><LF>Test message 002<CR><LF>
2018-05-14 22:49:59> IN: Input Protocol Simulator 'Input Simulator': [101]-"Test message 003"-MOMENTARY
2018-05-14 22:49:59> OUT: 'COMP2 Out' Encoded [101]-"Test message 003"-MOMENTARY
2018-05-14 22:49:59> OUT: Saving to Test Output File: 101<CR><LF>Test message 003<CR><LF>
2018-05-14 22:50:03> Updating System Configuration.
2018-05-14 22:50:03> Maximum Inputs and Outputs limited to 3.
2018-05-14 22:50:03> Examining Path #1.
2018-05-14 22:50:03> Path 1 Layer 1 Contains Block of Type 'input_protocol_simulator'.
2018-05-14 22:50:03> Path 1 Layer 2 Contains Block of Type 'output_protocol_comp2'.
2018-05-14 22:50:03> Path 1 Layer 2 Contains Block of Type 'test_output'.
2018-05-14 22:50:08> Updating System Configuration.
2018-05-14 22:50:08> Maximum Inputs and Outputs limited to 3.
```

Input Protocols

Click on the **Input Protocols** Toolbox item to display a list of available Input Protocol blocks. To use a block, you can scroll through the displayed blocks and either click a block or click and drag it onto the SNAP workspace.

Your system may not be licensed to allow use of all of the blocks defined in this document.

Input Simulator



The screenshot shows the configuration interface for the 'Input Simulator' block. It features a title bar 'Input Simulator' and a 'Toggle Simulator On/Off' checkbox. Below are several labeled input fields: 'Name' (Input Simulator), 'Base Test Message' (Test), 'Destination' (101), 'Repeat/Incr After (sec)' (10), and 'Max Alert Age (sec)' (600). A 'Modifiers' field is present but empty.

The Input Simulator block allows you to simulate the generation of alerts into the SNAP system, for purposes of testing your SNAP configuration. Once you configure the Input Simulator settings, you can click the Toggle Simulator On/Off checkbox to enable or disable the automatic generation of alerts.

A companion testing tool is the [Capture Output block](#).

The Input Simulator has the following elements:

- **Toggle Simulator On/Off checkbox** - Controls automatic generation of alerts
- **Name** - Used for activity logging
- **Base Test Message** - A sequential number is appended to this base test message value as each alert is generated
- **Destination** - The numeric or alphanumeric value that affects how alerts are routed through the SNAP system and through the systems connected to SNAP Output Protocols
- **Repeat/Incr After (sec)** - The time interval, in seconds, between each alert that is generated, ranging from 1 to 999. A numeric suffix value is incremented as each new alert is generated.
- **Max Alert Age (sec)** - The time interval, in seconds, that an active alert will transition to a canceled state. If the Max Alert Age value is 0 or blank, alert cancellation events will not be created. Allowable values for Max Alert Age include blank and 0 through 9999.

- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block

Plain Text Input Protocol

Plain Text Input <--- Input Method:

Name: Plain Text In

Cancel Word:

Include Input Name in Message

Default Destination: 101

Delimiter(s): <CR><LF>

Modifiers:

The Plain Text Input block allows processing of delimited text data into alerts, and requires use of one of the SNAP [Input Methods](#).

The Plain Text Input block has the following elements:

- [Input Method](#) - An Input Method is required to be snapped into this block in order for it to function
- **Name** - Used for activity logging and optionally being used as a prefix on the generated alert messages
- **Cancel Word** - A case-insensitive word value that, if non-blank, allows the Input Protocol block to detect if incoming data contains the Cancel Word as a means of being able to generate alert objects with a state value of "**canceled**". The cancel word can occur as either a prefix or a suffix on the incoming data, with or without leading or trailing space characters.
- **Include Input Name in Message checkbox** - If checked, the Name value will be used as a prefix on the message value of alerts generated in this block
- **Default Destination** - A numeric or alphanumeric Destination value that will be applied to all alerts generated by this block. You can use [Modifiers](#) to alter the Destination value as an alert object travels through the SNAP system.
- **Delimiter(s)** - One or more ASCII characters, which represent the end of each incoming Plain Text data packet, to trigger processing of the incoming data. The Delimiter values can be formatted using LT/GT methods or as readable text, such as +++. An example of allowable methods of formatting the typical carriage return/linefeed combo is:
<CR><LF>, <0x0D><0x0A>, or <13><10>
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block

Plain Text Input <--- Input Method: *Ethernet Input*

IP Port: 192.168.10.142

Name: Plain Text In

Cancel Word: cancel

Include Input Name in Message

Default Destination: 105

Delimiter(s): ---<CR>

Modifiers:

COMP2 Input Protocol

COMP2 Input <--- Input Method:

Name: COMP2 In

Cancel Word:

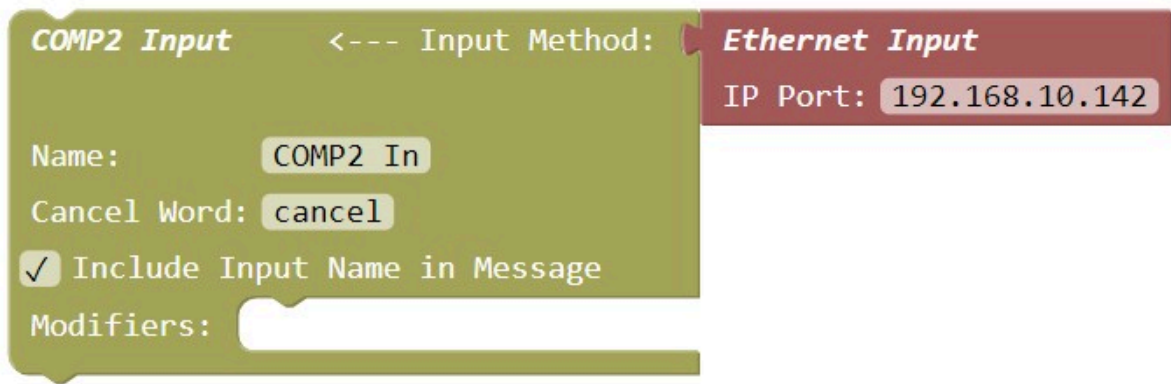
Include Input Name in Message

Modifiers:

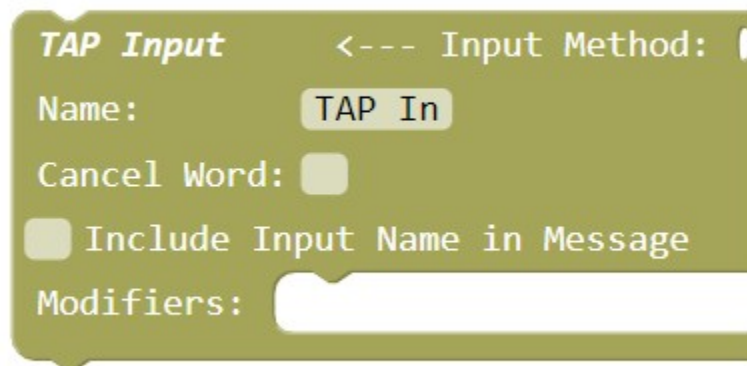
The COMP2 Input block allows processing of COMP2 protocol formatted data into alerts, and requires use of one of the SNAP [Input Methods](#). COMP2 protocol delivers both a Destination value and a Message value into the SNAP system.

The COMP2 Input block has the following elements:

- **Input Method** - An Input Method is required to be snapped into this block in order for it to function
- **Name** - Used for activity logging and optionally being used as a prefix on the generated alert messages
- **Cancel Word** - A case-insensitive word value that, if non-blank, allows the Input Protocol block to detect if incoming data contains the Cancel Word as a means of being able to generate alert objects with a state value of "canceled". The cancel word can occur as either a prefix or a suffix on the incoming data, with or without leading or trailing space characters.
- **Include Input Name in Message checkbox** - If checked, the Name value will be used as a prefix on the message value of alerts generated in this block
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block



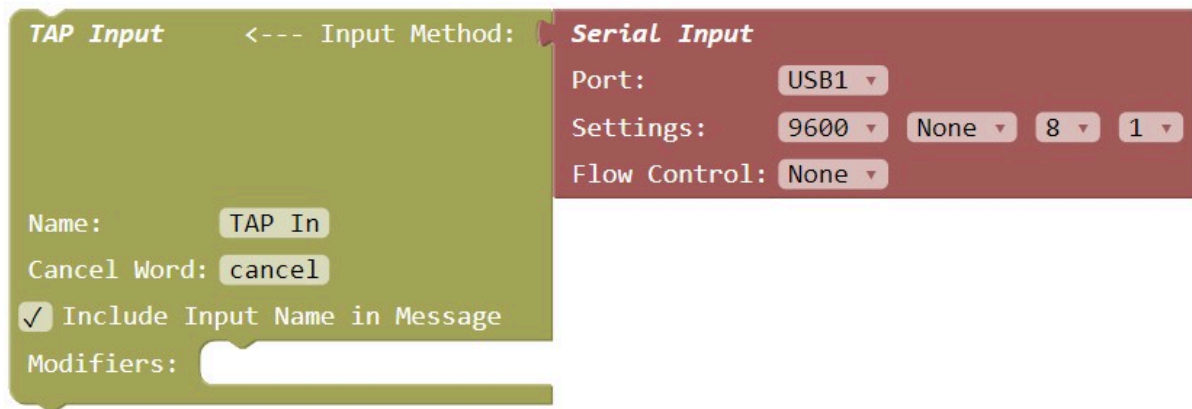
TAP Input Protocol



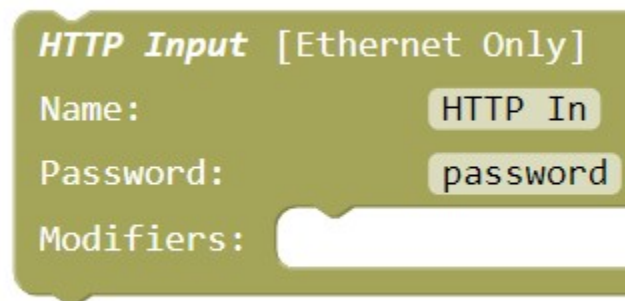
The TAP Input block allows processing of TAP protocol formatted data into alerts, and requires use of one of the SNAP [Input Methods](#). TAP protocol delivers both a Destination value and a Message value into the SNAP system.

The TAP Input block has the following elements:

- **Input Method** - An Input Method is required to be snapped into this block in order for it to function
- **Name** - Used for activity logging and optionally being used as a prefix on the generated alert messages
- **Cancel Word** - A case-insensitive word value that, if non-blank, allows the Input Protocol block to detect if incoming data contains the Cancel Word as a means of being able to generate alert objects with a state value of "**canceled**". The cancel word can occur as either a prefix or a suffix on the incoming data, with or without leading or trailing space characters.
- **Include Input Name in Message checkbox** - If checked, the Name value will be used as a prefix on the message value of alerts generated in this block
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block



HTTP Input Protocol



The HTTP Input block allows specialized HTTP Input server processing of secure HTTP GET or POST data into alerts, at the SNAP system sub URL value of **/api**, e.g. **192.168.10.142/api**. The HTTP Input protocol requires digest authentication as part of the connection process.

The HTTP Input block has the following elements:

- **Name** - Used for activity logging
- **Password** - The value required to be included as the password value combined with **api** as the username value in the digest access authentication process that is required for HTTP Input server connections. If the digest authentication fails or if the GET or POST parameters are malformed, the alert will not be processed.
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block

The HTTP Input protocol requires the following parameters:

- **destination** - The destination value to be assigned to the alert object created from the incoming data
- **message** - The message value to be assigned to the alert object created from the incoming data

- **state** - The state value to be assigned to the alert object created from the incoming data. Allowable state values include **active**, **cancel**, and **momentary**.

Following is an example HTTP Input URL, via the GET method, using the default HTTP Input configuration:

`http://api:password@192.168.10.142/api?destination=101&message=This%20is%20a%20test&state=active`

If the HTTP Input GET or POST attempt is successful, the HTTP Input server will respond with **ok**. Otherwise, the HTTP Input server will respond with **error**.

E-mail Input Protocol

E-mail Input (SMTP) [Ethernet Only]

Name:

IP Address/Domain:

Filter E-mail Address

Include Sender in Message

Include From: Header in Message

Include Subject in Message

Include Subject: Header in Message

Include Body in Message

Include Body: Header in Message

Modifiers:

E-mail Input (IMAP) [Ethernet Only]

Name:

IMAP Server:

IP Port:

Username:

Password:

Retrieval Interval (s):

Destination:

Include Sender in Message

Include From: Header in Message

Include Subject in Message

Include Subject: Header in Message

Include Body in Message

Include Body: Header in Message

Modifiers:

The E-mail Input block allows processing of either SMTP or IMAP protocol formatted data into alerts. You can use only one instance of E-mail Input, by choosing either SMTP or IMAP.

The E-mail Input SMTP service monitors IP Port 25 for incoming e-mail sent from SMTP client apps.

The E-mail Input IMAP service is configured to act as an IMAP client to an IMAP server. The IMAP client checks for new e-mail in the assigned IMAP mailbox using the Retrieval Interval value defined in the E-mail Input (IMAP) block. The IMAP client does not mark the e-mail as read, allowing a human to also interact with the same mailbox on the IMAP server. The IMAP client only monitors one mailbox, requiring the customer to send all alerts to a single e-mail address managed by the IMAP server.

The alert Destination value is extracted from the prefix of the e-mail To: and/or CC: address fields and the alert Message value is formatted using the E-mail Input Block configuration elements. As an example, a To: address value of 101@snap.com will generate a Destination value of "**101**".

Note that if incoming e-mail includes multiple e-mail addresses in the To: and/or CC: fields, all of those e-mail addresses will be processed

The E-mail Input (SMTP) block has the following elements:

- **Name** - Used for activity logging
- **IP Address/Domain** - An optional domain type of value that can be used with the Filter E-mail Address checkbox defined below. If IP Address/Domain value is blank, no e-mail address filtering will occur. This value represents the expected allowable right portion of the To: address on incoming e-mail, such as 192.168.10.142 or snap.com.
- **Filter E-mail Address checkbox** - If checked, and if the IP Address/Domain field value is non-blank, incoming e-mail will not be processed unless the To: or CC: field domain level values match the IP Address/Domain field value
- **Include Sender in Message checkbox** - If checked, the e-mail Sender value will be used as a part of the message value of alerts generated in this block
- **Include From: Header in Message checkbox** - If checked, the value "**From:**" will be used to identify the From: portion of the e-mail in the alert messages generated in this block
- **Include Subject in Message checkbox** - If checked, the e-mail Subject value will be used as a part of the message value of alerts generated in this block
- **Include Subject: Header in Message checkbox** - If checked, the value "**Subject:**" will be used to identify the Subject: portion of the e-mail in the alert messages generated in this block
- **Include Body in Message checkbox** - If checked, the e-mail Body value will be used as a part of the message value of alerts generated in this block
- **Include Body: Header in Message checkbox** - If checked, the value "**Body:**" will be used to identify the Body: portion of the e-mail in the alert messages generated in this block
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block

E-mail Input (IMAP) [Ethernet Only]

Name:

IMAP Server:

IP Port:

Username:

Password:

Retrieval Interval (s):

Destination:

Include Sender in Message

Include From: Header in Message

Include Subject in Message

Include Subject: Header in Message

Include Body in Message

Include Body: Header in Message

Modifiers:

The E-mail Input (IMAP) block has the following elements:

- **Name** - Used for activity logging
- **IMAP Server** - Defines the connection address of the IMAP server
- **IP Port** - Defines the connection IP port of the IMAP server
- **Username** - Defines the username of the mailbox to be monitored on the IMAP server
- **Password** - Defines the password of the mailbox to be monitored on the IMAP server
- **Retrieval Interval (s)** - Defines how often the mailbox will be checked on the IMAP server, in units of seconds
- **Destination** - Defines the Destination value to be as the destination value of alerts generated in this block
- **Include Sender in Message checkbox** - If checked, the e-mail Sender value will be used as a part of the message value of alerts generated in this block
- **Include From: Header in Message checkbox** - If checked, the value "**From:**" will be used to identify the From: portion of the e-mail in the alert messages generated in this block
- **Include Subject in Message checkbox** - If checked, the e-mail Subject value will be used as a part of the message value of alerts generated in this block
- **Include Subject: Header in Message checkbox** - If checked, the value "**Subject:**" will be used to identify the Subject: portion of the e-mail in the alert messages generated in this block
- **Include Body in Message checkbox** - If checked, the e-mail Body value will be used as a part of the message value of alerts generated in this block
- **Include Body: Header in Message checkbox** - If checked, the value "**Body:**" will be used to identify the Body: portion of the e-mail in the alert messages generated in this block
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block

Inovonics Input Protocol

Inovonics Input <--- Input Method: **Serial Input**

Port: USB3

Settings: 9600 None 8 1

Flow Control: None

Name: Inovonics In

Include Input Name in Message

Default Destination: 101

Maintenance Destination: 102

Report Destination:

Send Report At: Never

No Check In Delay: (min) 0

Use Custom Tamper Messages

Transmitters Receivers/Repeaters

Location Report

Network ID EN6040 Only

Modifiers:

The Inovonics Input block allows processing of Inovonics protocol formatted data into alerts, and typically requires use of the SNAP [Serial Input Method](#). Inovonics protocol provides alert activation and cancellation information, along with Transmitter address and nearest Receiver address information. The formatting of alert Destination and Message values is dependent upon configuring the [Inovonics Transmitters](#) and the [Receivers](#) tables. These alerts will have the Default Destination value applied to them if a Destination value is not assigned in the Transmitters table.

Note that the quantity of Inovonics transmitters that may be processed is limited by licensing, and that additional Inovonics transmitter processing capacity can be obtained by purchasing additional licenses.

Inovonics protocol also provides maintenance alert information, such as low battery, tamper, etc. Maintenance alerts will have the Maintenance Destination value applied to them.

Inovonics [Transmitters](#) and [Receivers/Repeaters](#) are auto-detected and configurable in tables. Note that Transmitters and Receivers/Repeaters are not enabled by default, and that you must click the Enabled checkbox in order for their signals to be processed by the system.

The Inovonics Input block has the following elements:

- **Input Method** - An Input Method is required to be snapped into this block in order for it to function. Typically the [Serial Input Method](#) is used, but with appropriate infrastructure, the Ethernet Input Method can be used. Inovonics requires Settings values of 9600 None 8 and 1.
- **Name** - Used for activity logging and optionally being used as a prefix on the generated alert messages
- **Include Input Name in Message checkbox** - If checked, the Name value will be used as a prefix on the message value of alerts generated in this block

- **Default Destination** - A numeric or alphanumeric Destination value that will be applied to all alerts generated by this block. You can use [Modifiers](#) to alter the Destination value as an alert object travels through the SNAP system.
- **Maintenance Destination** - A numeric or alphanumeric Destination value that will be applied to all maintenance alerts generated by this block, such as low battery or tamper. You can use [Modifiers](#) to alter the Destination value as an alert object travels through the SNAP system.
- **Report Destination** - An e-mail address for sending a daily e-mailed tracking report. For this feature to work, the E-mail Output feature needs to also be operational.
- **Send Report At selector** - A selector that allows selection of an hour value in which to send a daily e-mailed tracking report. The default value is Never. For this feature to work, the E-mail Output feature needs to also be operational.
- **No Check In Delay (min)** - A numeric value that allows adjustment of when the No Check In maintenance alert is generated for a particular Inovonics transmitter. The default No Check In timeout is 5 minutes for all transmitters other than the EN1221S senior pendant, which has a 100 minute No Check In timeout default value. The default No Check In Delay value is 0 and can range up to 999 minutes. If the No Check In Delay value is set to 5, as an example, the No Check In timeout in the system will be 10 minutes for most transmitters and 105 minutes for the EN1221S.
- **Use Custom Tamper Messages checkbox** - If checked, a non-blank Custom Tamper value in the [Inovonics Transmitters table](#) will be used as the alert message when a tamper event is detected for a given transmitter.
- **Transmitters button** - Clicking this button opens the [Inovonics Transmitters table](#)
- **Receivers/Repeaters button** - Clicking this button opens the [Inovonics Receivers table](#)
- **Location Report button** - Clicking this button opens the [Inovonics Location Report](#), which is a report of the last known location of all enabled Inovonics transmitters. The report dynamically updates while it is open. Red colored records indicate transmitter detection problems. Timestamp values use a 24 hour format.
- **Network ID button** - Clicking this button opens a page that allows you to edit a Network ID, which is a numeric value ranging from 1 to 31, which is used only by the Inovonics EN6040 Network Coordinator model of central receiver, to improve the data bandwidth of the Inovonics system in larger facilities, using directed messaging, and to isolate your network data from others. The Network ID is assigned to both the EN6040 central receiver and to all of the enabled repeaters in your network. Network ID must be distributed to all of the repeaters. To update a repeater's Network ID value, the repeater must be enabled in the [Receivers/Repeaters](#) table and the repeater must be reset at the repeater, after the Network ID value is edited. WARNING: If you configure your system Network ID throughout the system, then later change the Network ID value in the Inovonics input, you must reset each affected repeater in order for the system to operate correctly. Note that if you want to revert a previously programmed repeater to operate without directed messaging and to become compatible with the EN4000 and EN4200 central receivers, you must set the Network ID value to zero (0), and reset each repeater so that it changes its behavior to default with the new Network ID assignment.
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block

Inovonics Input <--- Input Method:

Name:

Include Input Name in Message

Default Destination:

Maintenance Destination:

Report Destination:

Send Report At:

No Check In Delay: (min)

Use Custom Tamper Messages

Network ID

Modifiers:

Serial Input

Port:

Settings:

Flow Control:

Close Refresh

Inovonics Transmitters

Filter :type any text here HELP

Address	Translation	Translation Secondary	Destination	Enabled	Include Location	Model	Momentary	Invert Primary	Invert Secondary	Invert Tamper	Custom Tamper	Delete
10729611	10729611			<input type="checkbox"/>	<input type="checkbox"/>	EN1210	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
10619391	10619391			<input type="checkbox"/>	<input type="checkbox"/>	EN1210	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
09304120	Primary	Secondary		<input checked="" type="checkbox"/>	<input type="checkbox"/>	EN1212	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Custom Tamper	
07367366	07367366			<input type="checkbox"/>	<input type="checkbox"/>	E*1223S	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
10619399	10619399			<input type="checkbox"/>	<input type="checkbox"/>	EN1210	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
07367382	07367382			<input type="checkbox"/>	<input type="checkbox"/>	E*1223S	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
11144690	11144690			<input type="checkbox"/>	<input type="checkbox"/>	EN1221S-60N/W	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
11190290	11190290			<input type="checkbox"/>	<input type="checkbox"/>	EN1221S-60N/W	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
11169183	11169183			<input type="checkbox"/>	<input type="checkbox"/>	EN1221S-60N/W	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
08756818	08756818			<input type="checkbox"/>	<input type="checkbox"/>	EN1221S-60N/W	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
11169171	11169171			<input type="checkbox"/>	<input type="checkbox"/>	EN1221S-60N/W	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

The Inovonics Transmitters table displays the auto-populated list of detected Inovonics transmitters



The Inovonics Transmitters table has the following elements:

- **Refresh** - Clicking Refresh causes the table to be reloaded, and is typically used when trying to introduce a new transmitter into your system.
- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Address** - A non-editable reference Address value for the Transmitter
- **Translation** - The Message value that you want to be used for alerts generated by the selected Transmitter
- **Translation Secondary** - The Message value that you want to be used for alerts generated by any secondary input on the selected Transmitter. On door/window transmitters, the primary input is treated

as the reed switch input. If the Translation Secondary field value is blank, no alerts will be generated for the associated transmitter's secondary input.

- **Destination** - The Destination value that you want to be used for alerts generated by the selected Transmitter. If you leave this field blank, the Default Destination value will be used.
- **Enabled** - If checked, alerts from this transmitter will be processed. By default, newly detected transmitters will be disabled.
- **Include Location** - If checked, the name of the nearest receiver will be included in the alert message
- **Model** - A non-editable reference value for the Transmitter Model, to help identify characteristics of Transmitter behavior
- **Momentary** - If checked, alerts from this transmitter will be generated with a Momentary [state](#)
- **Invert Primary** - If checked, signal detection for the primary alert will be inverted from normal. This is primarily used to accommodate transmitter polarity jumper settings and special use products.
- **Invert Secondary** - If checked, signal detection for the secondary alert will be inverted from normal. This is primarily used to accommodate transmitter polarity jumper settings and special use products.
- **Invert Tamper** - If checked, signal detection for the tamper alert will be inverted from normal. This is primarily used to accommodate transmitter polarity jumper settings and special use products.
- **Custom Tamper** - The alert Message value that you want to be used for tamper alerts generated by the selected Transmitter. If you leave this field blank, the Default Tamper alert message value will be used.
- **Delete** - Allows deletion of the selected table record

Close

Inovonics Receivers				Refresh
Filter :type any text here			HELP	
Address	Name	Enabled	Delete	
12639144	Central	<input checked="" type="checkbox"/>		
06252738	East Wing	<input checked="" type="checkbox"/>		
12649847	West Wing	<input checked="" type="checkbox"/>		

The Inovonics Receivers table displays the auto-populated list of detected Inovonics receivers and repeaters

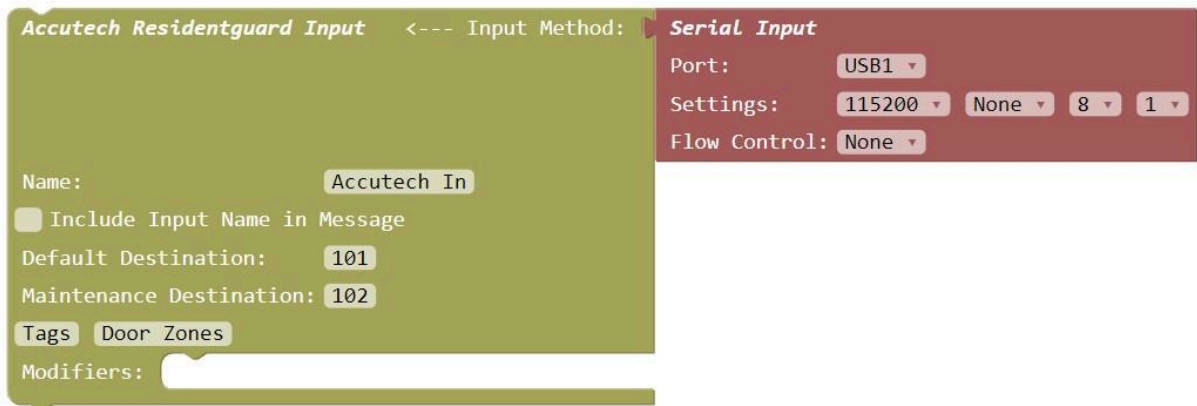
The Inovonics Receivers table has the following elements:

- **Refresh** - Clicking Refresh causes the table to be reloaded, and is typically used when trying to introduce a new receiver into your system.
- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Address** - A non-editable reference Address value for the Receiver
- **Name** - The value that will be included in the alert message when Include Location is enabled in the [Transmitters table](#)
- **Enabled** - If checked, alerts reported by this receiver will be processed. By default, newly detected receivers will be disabled.
- **Delete** - Allows deletion of the selected table record

Inovonics Location Report

Transmitter	Nearest Receiver	Last Seen
09304120	No Data Yet	N/A
13082963	No Data Yet	N/A
gotcha	EN6040	17:55:30 Dec 09 2019

Accutech Input Protocol



Accutech Residentguard Input <--- Input Method: **Serial Input**

Port: USB1

Settings: 115200 None 8 1

Flow Control: None

Name: Accutech In

Include Input Name in Message

Default Destination: 101

Maintenance Destination: 102

Tags Door Zones

Modifiers:

The Accutech Residentguard Input block allows processing of Accutech Residentguard protocol formatted data into alerts, and typically requires use of the SNAP [Serial Input Method](#). Accutech protocol provides alert activation and cancellation information, along with Tag address and Door Zone address information. The formatting of alert Destination and Message values is dependent upon configuring the [Tags](#) and the [Door Zones](#) tables. These alerts will have the Default Destination value applied to them if a Destination value is not assigned in the Tags table.

Accutech protocol also provides maintenance alert information, such as low battery, tamper, etc. Maintenance alerts will have the Maintenance Destination value applied to them.

Accutech [Tags](#) and [Door Zones](#) are auto-detected and configurable in tables

The Accutech Input block has the following elements:




- **Input Method** - An Input Method is required to be snapped into this block in order for it to function. Typically the [Serial Input Method](#) is used, but with appropriate infrastructure, the Ethernet Input Method can be used. Accutech requires Settings values of 115200 None 8 and 1.
- **Name** - Used for activity logging and optionally being used as a prefix on the generated alert messages
- **Include Input Name in Message checkbox** - If checked, the Name value will be used as a prefix on the message value of alerts generated in this block
- **Default Destination** - A numeric or alphanumeric Destination value that will be applied to all alerts generated by this block. You can use [Modifiers](#) to alter the Destination value as an alert object travels through the SNAP system.

- **Maintenance Destination** - A numeric or alphanumeric Destination value that will be applied to all maintenance alerts generated by this block, such as low battery or tamper. You can use [Modifiers](#) to alter the Destination value as an alert object travels through the SNAP system.
- **Tags button** - Clicking this button opens the [Accutech Tags table](#)
- **Door Zones button** - Clicking this button opens the [Accutech Door Zones table](#)
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block

[Close](#)

Accutech Tags [Refresh](#)

Filter :type any text here [+ Add New Accutech Tag](#) [HELP](#)

Address	Name	Destination	Delete
5	R. Jones		
1	C. Smith		
6	L. Lewis		

The Accutech Tags table displays the auto-populated list of detected Accutech tags


The Accutech Tags table has the following elements:

- **Refresh** - Clicking Refresh causes the table to be reloaded, and is typically used when trying to introduce a new tag into your system.
- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Accutech Tag button** - Clicking this button allows you to add a new Accutech Tag record. Note that the Accutech Tags table is auto-populated, but you can choose to manually add Accutech Tag records.
- **Address** - A non-editable reference Address value for the Tag
- **Name** - The Tag reference value that you want to be used for alerts generated in reference to the selected Tag
- **Destination** - The Destination value that you want to be used for alerts generated in reference to the selected Tag. If you leave this field blank, the Default Destination value will be used.
- **Delete** - Allows deletion of the selected table record

[Close](#)

Accutech Door Zones [Refresh](#)

Filter :type any text here [HELP](#)

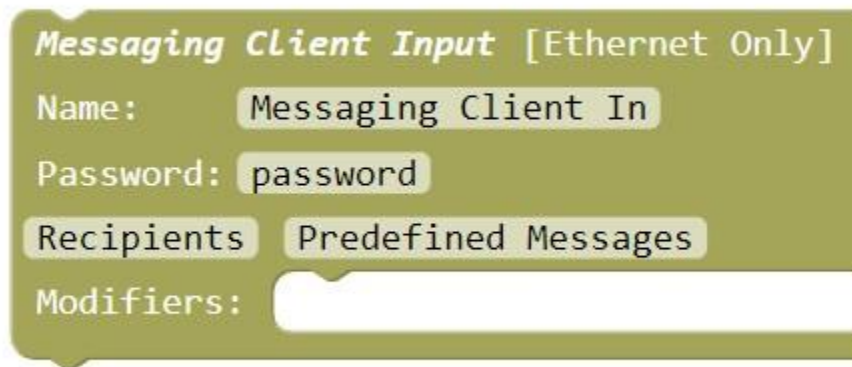
Address	Name	Delete
30	Lobby Door	

The Accutech Door Zones table displays the auto-populated list of detected Accutech Door Zones

The Accutech Door Zones table has the following elements:

- **Refresh** - Clicking Refresh causes the table to be reloaded, and is typically used when trying to introduce a new door zone into your system.
- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Address** - A non-editable reference Address value for the Door Zone
- **Name** - The value that will be included in the alert message when Door Zone related alerts are detected
- **Delete** - Allows deletion of the selected table record

Messaging Client Input Protocol



The screenshot shows a configuration window titled "Messaging Client Input [Ethernet Only]". It contains several input fields and buttons:

- Name:** Messaging Client In
- Password:** password
- Recipients:** Predefined Messages
- Modifiers:** (empty field)

The Messaging Client Input block activates processing of browser based messaging to selected alert recipients. Alert Destination values are configured in the [Recipients](#) table.

To view the Messaging Client page, log into the **/messaging** sub-URL of SNAP, e.g. **192.168.10.142/messaging**, using the **messaging** username value and the Password value defined in the Messaging Client Input block. Alternatively, you can log into the SNAP primary URL with a username value of messaging, to cause the Messaging Client page to appear.

You can configure desktop shortcut URLs to automatically connect and log into the Messaging Client page of the SNAP server using the required login credentials. Following is an example URL:

192.168.10.312/messaging?username=messaging&password=password

The Messaging Client Input block has the following elements:

- **Name** - Used for activity logging
- **Password** - The site wide password required when logging into SNAP using the **messaging** userid value
- **Recipients button** - Clicking this button opens the [Messaging Recipients table](#)
- **Predefined Messages button** - Clicking this button opens the [Predefined Messages table](#)
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block



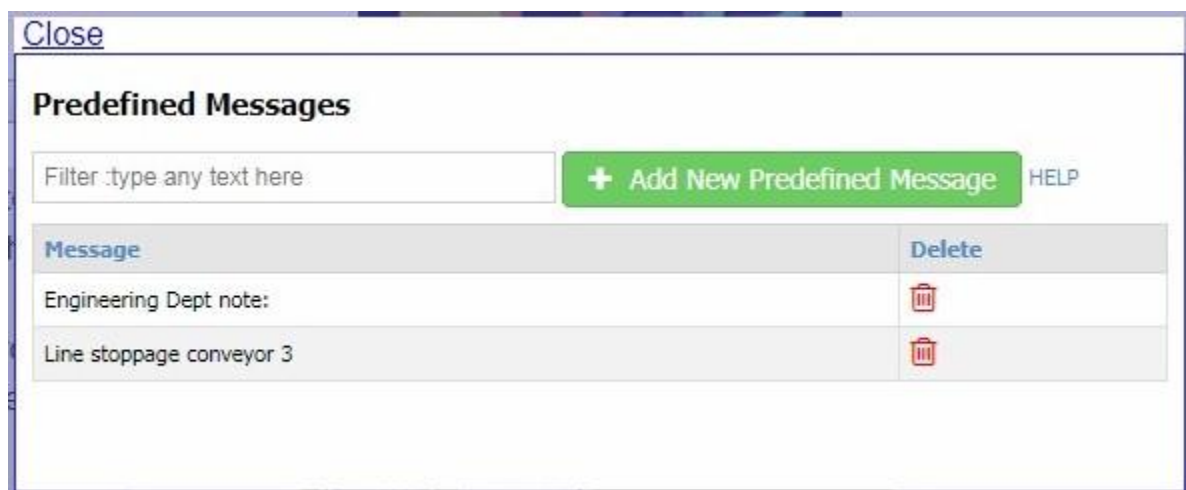
The Messaging Recipients table displays a list of recipient names that you want to appear in the [SNAP Messaging page](#).

To define messaging groups, reuse Name values while building the Messaging Recipients table. When the Recipients list in the [SNAP Messaging page](#) is rendered, Messaging Recipients table records with duplicate Name values are consolidated as a single selection in the SNAP Messaging Page Recipients list. In this example, Engineering is defined as a recipient name value that is processed as a group.

When using the SNAP Messaging Page, and a Recipient that is defined as a group is selected, the message will be delivered to all Destination values defined in the Messaging Recipients table that are associated with the selected Recipient. In this example, selecting Engineering will cause output to destination values of 212 and 213.

The Messaging Recipients table has the following elements:

- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Recipient** - Prompts for the Name of a new Messaging Recipients table record
- **Name** - The Messaging Recipient value used in the Recipients list display in the [SNAP Messaging page](#)



The Predefined Messages table displays a list of predefined messages selectable in the [SNAP Messaging page](#)

The Predefined Messages table has the following elements:

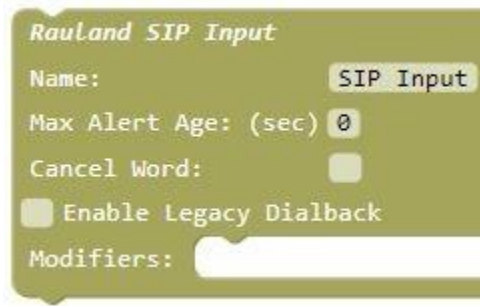
- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Predefined Message** - Prompts for the Message value of a new Predefined Messages table record
- **Message** - The value used in the Predefined Messages list display in the [SNAP Messaging page](#)
- **Delete** - Allows deletion of the selected table record

The screenshot displays the 'SNAP Messaging' interface. It features a 'Recipients' list on the left with items: Alex, Curtis, Dick, Engineering Dept (highlighted), Jackie, and Sales Dept. To the right is a 'Message to Send' text area containing 'Meeting reminder, 10:15, conf2'. Below this are 'Predefined Msgs', 'Time', and 'Date' dropdown menus. A 'Send' button and a 'clear' button are on the right. At the bottom, there is a 'Clear on Send' checkbox (checked), a 'Signature' field, and buttons for 'Edit Signature', 'Clear Signature', and 'Logout'. Below the form, there are three paragraphs of instructional text: 'Select one or more recipients (Click or Ctrl+Click), edit a message, then click SEND or Press the Enter key', 'You can load a predefined message by selecting from the Predefined Msgs list', and 'Message size is limited to 200 characters'. The final paragraph explains the signature and clear-on-send functionality.

The SNAP Messaging page allows multi-selection of Recipients and creation of messages to be delivered to the selected recipients. Note that in this example, the Recipients list contains both individual recipients and group recipients.

When using the SNAP Messaging Page, and a Recipient that is defined as a group is selected, the message will be delivered to all Destination values defined in the Messaging Recipients table that are associated with the selected Recipient. In this example, selecting Engineering Dept will cause output to destination values of 212 and 213, as defined in the [Messaging Recipients table](#).

Rauland SIP Input Protocol



The Rauland SIP Input block processes Rauland SIP protocol based alerts from Rauland systems. The SIP based alerts output by Rauland systems are designed to be processed by other SNAP elements to enhance the user experience.

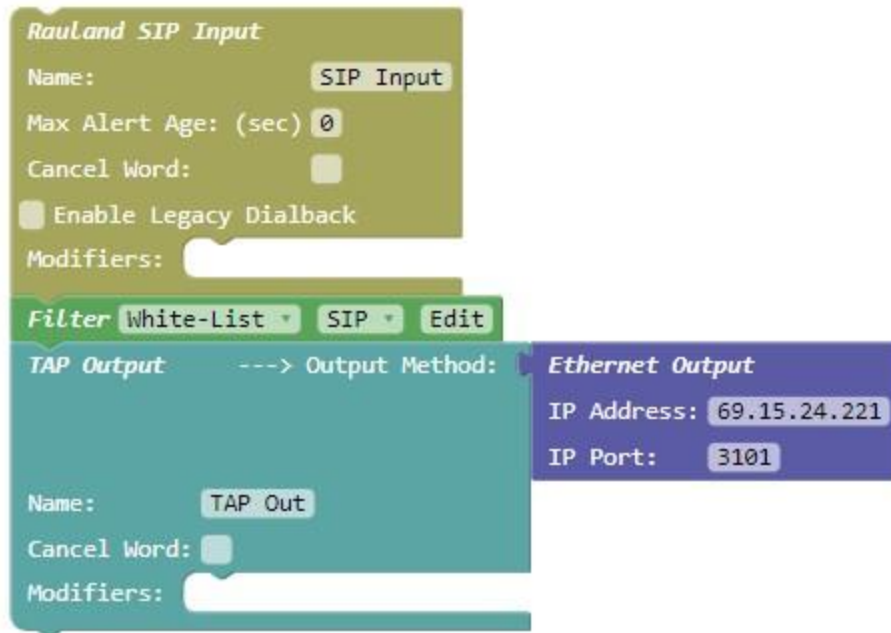
The Rauland SIP Input block will normally generate stateful alerts, e.g. initiating an alert with a **state** value of **active**. The only condition that stateless alerts with a state value of **momentary** will be created in the Rauland SIP Input block is when all of the following conditions are met:

- Cancel Word value in the Rauland SIP Input block is blank
- Max Alert Age value in the Rauland SIP Input block is zero (0)

The Rauland SIP Input block will generate alert objects with a state value of **canceled** under the following conditions:

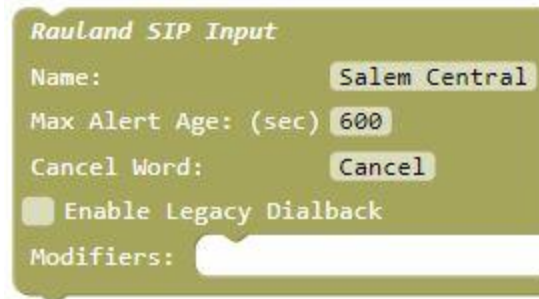
- A SIP notification message value contains the Cancel Word value defined in the Rauland SIP Input block
- The age of a given alert with a state value of **active** exceeds a non-zero Max Alert Age value in the Rauland SIP Input block
- A Rauland SIP notification is received that matches an active alert and includes a cancel signal

Common examples of Rauland SIP Input alert processing are when alerts received by the Rauland SIP Input block are processed by additional SNAP outputs, such as the TAP Output block. The following example configuration combines a [Filter Modifier](#) and a TAP Output to selectively output alerts processed by the Rauland SIP Input block. You could, as an example, set the Filter Modifier to **White-List** mode and configure it to allow alerts with a **destination** value of 200, for example, to be passed through for processing by the downstream blocks. This type of configuration allows you to emulate the WaveWare STG product feature set.



The Rauland SIP Input block has the following elements:

- **Name** - Used for activity logging
- **Max Alert Age (sec)** - An optional maximum age an active alert will be allowed to be in the **active** state in the SNAP system. If the Max Alert Age value is blank or 0, alerts received on the Rauland SIP Input will not ever be automatically changed to **canceled** state via a timer in the SNAP system. Allowable Max Alert Age values include blank, and 0 through 9999. The default value is 0. If either the Max Alert Age value is zero (0) or the Cancel Word value is blank, alerts will be created with **momentary** status. Otherwise, alerts will be created with an **active** status, to allow stateful tracking and cancellation of alerts.
- **Cancel Word** - A case-insensitive word value that, if non-blank, allows the Input Protocol block to detect if incoming data contains the Cancel Word as a means of being able to generate alert objects with a state value of "**canceled**". The cancel word can occur as either a prefix or a suffix on the incoming data, with or without leading or trailing space characters. The default value is blank. If either the Cancel Word value is blank or the Max Alert Age value is zero (0), alerts will be created with **momentary** status. Otherwise, alerts will be created with an **active** status, to allow stateful tracking and cancellation of alerts.
- **Enable Legacy Dial Back** - The Legacy Dial Back setting causes formatting of all alert message values using the Rauland Legacy Dial Back method, when the upstream Rauland system is configured to output alerts with Legacy Dial Back formatting. Following is an example showing the difference between the normal and Legacy Dial Back methods of formatting alert message values. Note that the Legacy Dial Back message formatting rule is that the first "word" of the normally formatted alert message is replaced by the asterisk delimited legacy dial back "word" containing area/room/bed information. Word is defined as any information that is followed by a space character.
4West 402 1 Patient OT is an example normal alert message formatting with space delimiters between area/room/bed
4*402*1 402 1 Patient OT is Legacy Dial Back alert message formatting of the above message
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block



Rauland SIP Input use cases are discussed below:

- **Default behavior** - All alerts detected by the Rauland SIP Input block will have their state defined by the Max Alert Age and Cancel Word settings. If alerts are generated with an **active** state, detection of cancel keywords will generate associated alerts with a **canceled** state.
- **Filter Modifier used** - A Filter Modifier block can filter out all or part of the alerts detected in the Rauland SIP Input block. The Filter Modifier would be able to allow SIP calls that are configured in Rauland to be non-answerable, to be filtered using destination and/or keyword value based rules.

SIP Server Dialplan Configuration

The simplest Dialplan Rule to use with a SIP Server is:

- Matching Pattern:

```
$request = ^INVITE
$registered = ^false
To = sip:(.+)
```

- Deploy Pattern:

```
To = sip:%1@{STG IP}
```

Where {STG IP} is the STG IP address. Following is a screenshot of a Brekeke SIP Server Dialplan configuration.



This configuration will forward all unregistered sip invites to the STG.

The matching pattern "To = sip:(.+)" can be changed to use a different regular expression. For example, "To = sip:(10[0-9])@" would match extensions 100-109. "To = sip:(999999)@" would forward only invites from extension 999999.

Input Methods

The Input Method blocks are snapped into appropriate Input Protocol blocks, to define the data collection physical interface required for the selected block.

Ethernet Input Method



The Ethernet Input Method block is snapped into appropriate Input Protocol blocks, to define an Ethernet method of data collection. Edit the IP Port value appropriate to the associated Input Protocol block, and the SNAP server will listen for Ethernet connections on the assigned IP port. Note that the IP Port value of 9090 is not allowed, as it is reserved for use with the JSON Output.



Serial Input Method

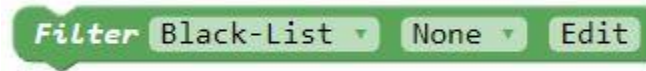


The Serial Input Method block is snapped into appropriate Input Protocol blocks, to define a serial port method of data collection. Edit the Port, Settings and Flow Control values appropriate to the associated Input Protocol block, and the SNAP server will listen for serial data at the assigned settings.

Modifiers

Modifier blocks can be snapped either between Protocol blocks, or in the Modifiers section of a Protocol block. When snapped between Protocol blocks, the Modifier block affects all SNAP alerts from all Protocol blocks and Modifier blocks above the Modifier block. When snapped into the Modifiers section of a Protocol block, the Modifier block only affects the SNAP alerts generated by the associated Protocol block.

Filter Modifier



The Filter Modifier block performs the process of filtering out SNAP alerts as they pass through the SNAP system, using white-list/black-list methods and rules defined in an assigned [Filter table](#).

You can create multiple Filter tables, so that you can deploy the Filter Modifier block in multiple places in your SNAP configuration structure, with each Filter Modifier having a unique Filter table assigned to it.

The Filter Modifier Block is composed of the following elements:

- **Black-List/White-List selector** - Black-List removes all alerts that match a record in the assigned Filter table. White-List removes all alerts except for those alerts that match a record in the assigned Filter table. The benefit of being able to choose White-List or Black-List is that it minimizes the amount of configuration required to achieve a particular filtering criteria. For a given Filter Modifier application, if you want all but a few alerts filtered out, then choose the White-List method. If you want only a few alerts filtered out, then choose the Black-List method.
- **Filter Table selector** - After creating one or more Filter tables, select a Filter Table from the list, to cause filtering processes to be applied to the selected table
- **Edit button** - Causes the Filter Tables table to appear, allowing the creation of one or more Filter tables.



Close

Filter Tables

Filter :type any text here + Add New Filter Table HELP

Name	Edit	Delete ▲
My Filter	Click Here to Open	

The Filter Tables table allows creation of multiple Filter Tables. The Filter Tables table has the following elements:




- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Filter Table** - Prompts for the Name value of a new Filter Tables table record

- **Name** - The value used in the Filter Table Selector of the [Filter Modifier Block](#)
- **Edit** - Click the link to open the associated Filter Table for editing
- **Delete** - Allows deletion of the selected table record

[Close](#)

Filter Table: My Filter [BACK](#)

Filter :type any text here [+ Add New Filter Record](#) [HELP](#)

Key String	Destination	Schedule	Delete ▾
checkin		Checkin	
	101	None	
code		None	

The Filter table allows editing of the selected Filter Table. The Filter table has the following elements:

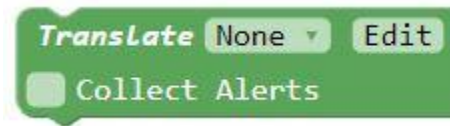
- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Filter Record** - Creates a blank Filter table record
- **Key String** - Optional key string value that is compared to alert message values in alerts processed by the Filter Modifier
- **Destination** - Optional destination value that is compared to alert destination values in alerts processed by the Filter Modifier
- **Schedule** - Optional schedule assignment to allow processing alerts for this Filter table record only when the assigned [Schedule](#) is active
- **Delete** - Allows deletion of the selected table record

The Filter Modifier uses the following filtering rules:

- Incoming alerts are compared to the all of the records in the assigned Filter table
- Key String comparisons to alert message values are case-insensitive, as are Destination comparisons
- If Black-List method is assigned in the Filter block, all alerts that match a record in the assigned Filter table will be removed.
- If White-List method is assigned in the Filter block, all alerts will be removed unless they match a record in the assigned Filter table.
- Filter table processing will fail to occur for a Filter table record under any of the following conditions:
 - Both Key String value is blank and Destination value is blank
 - If a Schedule is assigned and that Schedule is currently inactive
- Any non-blank Key String value is compared with an alert message value to see if the Key String is contained within the message, which defines a match between the alert and the Key String
- Any non-blank Destination value is compared with an alert destination value to determine if they match
- When both Key String and Destination values are non-blank in the Filter table record, incoming alerts must match both the Key String and Destination in order for a match to be defined
- When a schedule is assigned to a Filter table record and the schedule is currently inactive, that filter table record will not be processed

In the above Filter table example, the first record has both a Key String and Schedule defined, which causes alerts to be compared to the Key String value when the assigned schedule is active. The second record has only a Destination value defined, which causes alerts at all times to have their destination value compared to this Destination value. The third row has only a Key String value defined, which causes alerts at all times to have their message value examined to see if it contains the assigned Key String value.

Translate Modifier

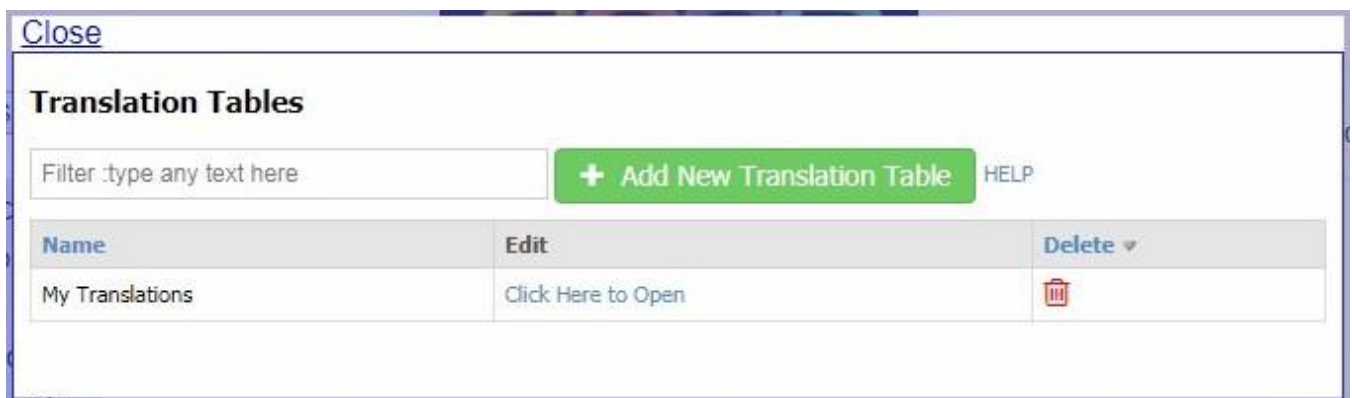
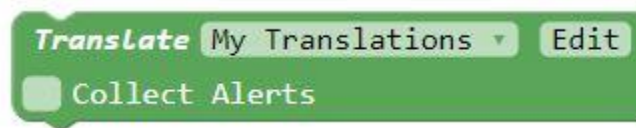


The Translate Modifier block performs the process of translating alert message values as they pass through the SNAP system, using rules defined in an assigned [Translation table](#).

You can create multiple Translation tables, so that you can deploy the Translate Modifier block in multiple places in your SNAP configuration structure, with each Translate Modifier having a unique Translation table assigned to it.

The Translate Modifier Block is composed of the following elements:

- **Translation Table selector** - After creating one or more Translation tables, select a Translation Table from the list, to cause translation processes to be applied to the selected table
- **Edit button** - Causes the Translation Tables table to appear, allowing the creation of one or more Translation tables.
- **Collect Alerts checkbox** - If checked, incoming alert objects are auto-populated into the assigned Translation Table



The Translation Tables table allows creation of multiple Translation Tables. The Translation Tables table has the following elements:




- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Translation Table** - Prompts for the Name value of a new Translation Tables table record

- **Name** - The value used in the Translation Table Selector of the [Translate Modifier Block](#)
- **Edit** - Click the link to open the associated Translation Table for editing
- **Delete** - Allows deletion of the selected table record

Close

Translation Table: My Translations Refresh BACK

Filter :type any text here + Add New Translation Record [HELP](#)

Key String	Destination	Replacement String	Replace Entire Message	Delete
code		Event	<input type="checkbox"/>	
	201	not very useful	<input type="checkbox"/>	
alarm 7	203	Sprinkler 7 failure	<input checked="" type="checkbox"/>	

The Translation table allows editing of the selected Translation Table. If the Collect Alerts checkbox is checked on the [Translate Modifier Block](#), and this table is selected in the Translate block's Filter Table selector, then this table will be able to create new table records automatically as new alerts are detected, where the combination of message value and destination value are compared to the Translation table's Key String and Destination fields to determine if a new record should be created.

The Translation table has the following elements:

- **Refresh** - Clicking Refresh causes the table to be reloaded, and is typically used when the Collect Alerts mode is active, to see if any new alerts have been detected and added to the table
- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Translation Record** - Creates a blank Translation table record
- **Key String** - Optional key string value that is compared to alert message values in alerts processed by the Translate Modifier
- **Destination** - Optional destination value that is compared to alert destination values in alerts processed by the Translate Modifier
- **Replacement String** - Optional string value that is used to either replace a matching key string in an alert message or to replace the entire alert message if the Key String value is blank
- **Replace Entire Message checkbox** - If checked, the entire alert message value will be replaced by the Replacement String value when an alert matches a record in an assigned Translation table
- **Delete** - Allows deletion of the selected table record

The Translate Modifier uses the following translation rules:

- Incoming alerts are compared to the all of the records in the assigned Translation table, unless a matching table record causes replacement of the entire alert message value
- Key String comparisons to alert message values are case-insensitive, as are Destination comparisons
- Translation table processing will fail to occur for a Translation table record if both the Key String value and Destination value are blank
- Any non-blank Key String value is compared with an alert message value to see if the Key String is contained within the message, which defines a match between the alert and the Key String

- Any non-blank Destination value is compared with an alert destination value to determine if they match
- When both Key String and Destination values are non-blank in the Translation table record, incoming alerts must match both the Key String and Destination in order for a match to be defined
- If Destination value is blank, Key String value is contained in the alert message, and Replace Entire Message is checked, the entire alert message will be replaced by the Replacement String value, and further translation processing will be stopped for that alert
- If Destination value is blank, Key String value is contained in the alert message, and Replace Entire Message is unchecked, if the Replacement String value is non-blank, only the portion of the alert message that matches the Key String will be replaced by the Replacement String value. In this case, if the Replacement String value is blank, then the portion of the alert message that matches the Key String will be removed.
- If the Key String value is blank, the Destination value is non-blank, and the alert destination value matches the Destination value of the table record, the entire alert message will be replaced by the Replacement String value, and further translation processing will be stopped for that alert
- If both the Key String and Destination table record values are non-blank, and an alert matches both, if Replace Entire Message is checked, the entire alert message will be replaced by the Replacement String value, and further translation processing will be stopped for that alert. In this case, if the Replace Entire Message checkbox is unchecked, then only the portion of the alert message that matches the Key String will be replaced by the Replacement String value.
- If any matching table record causes replacement of the entire alert message value, no more Translation table records will be examined for matches. Otherwise, if only the portion of an alert message value that matches a Key String value is replaced, then subsequent table records will be examined for further translation opportunities.
- When the Collect Alerts mode is active, each new combination of alert message and alert destination will cause creation of a new table record. However, if an existing table record has a non-blank Key String value and a blank Destination value, any alert whose message contains the Key String value will be considered a match and will not cause creation of a new table record. In order to minimize table record creation when performing the Collect Alerts process, for conditions where the same alert is sent to multiple Destinations, you can make the Destination field blank in one of those table records, to stop the creation of new records containing the same Key String value.

In the above Translation table example, the first record causes the portion of the alert message that contains the Key String value of **code** to be replaced by the Replacement String value of **Event**. The second record has only a Destination value defined, which causes any alerts with matching destination value to have their message value replaced by the Replacement String value of **not very useful**. The third row has both a Key String value and Destination value defined, in addition to Replace Entire Message checked, which causes alerts whose message contains the Key String value of **alarm 7** and have a destination value of **203** to have their message value replaced by the Replacement String value of **Sprinkler 7 failure**.

If you want to create a heartbeat signal in the SNAP for use by a downstream watchdog, you can combine the Input Simulator with the Translate Modifier, as shown below, where the Translate Modifier can be configured to strip the sequential message numbers added by the Input Simulator.

Input Simulator

Toggle Simulator On/Off

Name: Watchdog

Base Test Message: Heartbeat

Destination: 101

Repeat/Incr After (sec): 60

Max Alert Age (sec): 0


Modifiers: Translate Heartbeat Edit

Collect Alerts

Close

Translation Table: Heartbeat Refresh BACK

Filter :type any text here + Add New Translation Record HELP

Key String	Destination	Replacement String	Replace Entire Message	Delete
Heartbeat		Heartbeat	<input checked="" type="checkbox"/>	

Route Modifier

Route None Edit

The Route Modifier block performs the process of routing and duplicating alerts via their destination values as they pass through the SNAP system, using rules defined in an assigned [Route table](#).

You can create multiple Route tables, so that you can deploy the Route Modifier block in multiple places in your SNAP configuration structure, with each Route Modifier having a unique Route table assigned to it.


The Route Modifier Block is composed of the following elements:

- **Route Table selector** - After creating one or more Route tables, select a Route Table from the list, to cause routing processes to be applied to the selected table
- **Edit button** - Causes the Route Tables table to appear, allowing the creation of one or more Route tables.

[Close](#)

Route Tables

Filter :type any text here + Add New Route Table HELP

Name	Edit	Delete
My Routes	Click Here to Open	

The Route Tables table allows creation of multiple Route Tables. The Route Tables table has the following elements:




- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Route Table** - Prompts for the Name value of a new Route Tables table record
- **Name** - The value used in the Route Table Selector of the [Route Modifier Block](#)
- **Edit** - Click the link to open the associated Route Table for editing
- **Delete** - Allows deletion of the selected table record

[Close](#)

Route Table: My Routes

[BACK](#)

Filter :type any text here + Add New Route Record HELP

Key String	Old Destination	New Destination	Delete
	101	101	
	101	202@gmail.com	
test		203	
	101	201	

The Route table allows editing of the selected Route Table. The Route table has the following elements:

- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Route Record** - Creates a blank Route table record
- **Key String** - Optional key string value that is compared to alert message values in alerts processed by the Route Modifier
- **Old Destination** - Optional destination value that is compared to alert destination values in alerts processed by the Route Modifier
- **New Destination** - String value that is used to define the destination value of an alert matching the Route table record
- **Delete** - Allows deletion of the selected table record

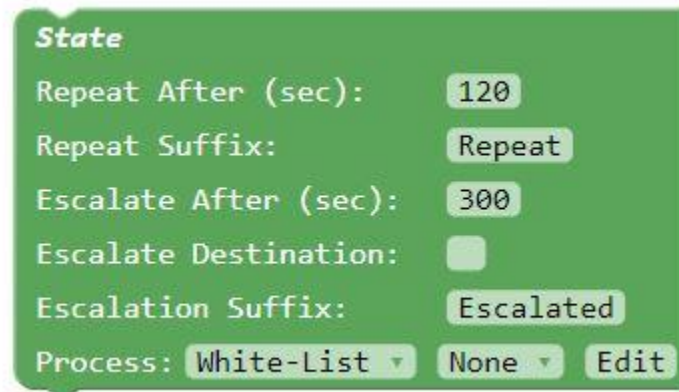
The Route Modifier uses the following routing rules:

- Incoming alerts are compared to the all of the records in the assigned Route table
- Key String comparisons to alert message values are case-insensitive, as are Destination comparisons

- Route table processing will fail to occur for a Route table record under the following conditions:
 - If both the Key String value and Old Destination value are blank
 - If the New Destination value is blank
- Any non-blank Key String value is compared with an alert message value to see if the Key String is contained within the message, which defines a match between the alert and the Key String
- Any non-blank Old Destination value is compared with an alert destination value to determine if they match
- When both Key String and Old Destination values are non-blank in the Route table record, incoming alerts must match both the Key String and Destination in order for a match to be defined
- When an incoming alert does not match any records in the Route table, the alert is passed through the SNAP system unaffected.
- When an incoming alert does match one or more records in the Route table, the original alert is not passed through the SNAP system, but a new alert is created for each matching record, retaining the original alert message value and assigned a destination value defined by the New Destination value of the matching Route table record.
- To allow an alert to be routed to both its original destination and to one or more new destinations, one Route table record should have matching Old Destination and New Destination values, and other records would repeat the same Old Destination value but would have new values in their New Destination field
- If Old Destination value is blank and Key String value is contained in the alert message, the affected alert will retain its message value and be assigned a new destination value defined by the New Destination value of the affected Route table record
- If the Key String value is blank, the Old Destination value is non-blank, and the alert destination value matches the Old Destination value, the affected alert will retain its message value and be assigned a new destination value defined by the New Destination value of the affected Route table record

In the above Route table example, the first, second, and fourth records cause matching alerts to be split into three downstream alerts, with one alert retaining its original destination value of **101**, while the other two alerts are assigned destination values of **202@gmail.com** and **201**. The third record causes any alert with a message value containing the Key String value of **test** to have its destination value changed to the New Destination value of **203**. Note that any alerts not containing **test** in their message value and not having a destination value of **101** will not be affected by this Route table and will be passed downstream in the SNAP processing system.

State Modifier



The image shows a configuration panel for the State Modifier. It has a green background and contains the following fields:

- State** (header)
- Repeat After (sec):** 120
- Repeat Suffix:** Repeat
- Escalate After (sec):** 300
- Escalate Destination:** (empty)
- Escalation Suffix:** Escalated
- Process:** White-List (dropdown), None (dropdown), Edit (button)

The State Modifier block performs the process of repeating and escalating alerts based upon how the State Modifier block is configured. The State Modifier can generate alerts to both the original alert destination and the Escalation Destination. The State Modifier is designed to be applied to SNAP alert objects that go through both the [active and cancel states](#). The alert object cancel state is necessary for the State Modifier to know when to stop generating repeat and escalation alerts for a given original alert. Alerts with a [state value of momentary](#) will not be processed by the State Modifier.

The State Modifier can be used multiple times in a system configuration, but only makes sense if you apply a [Filter table](#) for processing, where each filter table causes processing for different alert destination values. An example might be the need to repeat maintenance alerts on a different repeat interval than other types of alerts.

The State Modifier Block is composed of the following elements:

- **Repeat After (sec)** - A time interval, in seconds, after an alert is first detected and a related alert cancellation event has not been detected yet, at which a repeat alert will be created. If the value is blank or 0, repeat alerts will not be created. Allowable values include blank and 0 through 9999.
- **Repeat Suffix** - A string value that will be added as a suffix to any generated repeat alerts
- **Escalate After (sec)** - A time interval, in seconds, after an alert is first detected and a related alert cancellation event has not been detected yet, at which an escalation alert will be created. If the value is blank or 0, escalation alerts will not be created. Allowable values include blank and 0 through 9999.
- **Escalate Destination** - A string value that will be added as a suffix to any generated escalation alerts
- **Escalation Suffix** - Causes the Route Tables table to appear, allowing the creation of one or more Route tables.
- **Black-List/White-List selector** - Black-List processes all alerts other than those that match a record in the assigned Filter table. White-List processes only those alerts that match a record in the assigned Filter table. The benefit of being able to choose White-List or Black-List is that it minimizes the amount of configuration required to achieve a particular filtering criteria. For a given Filter Modifier application, if you want all but a few alerts filtered out, then choose the White-List method. If you want only a few alerts filtered out, then choose the Black-List method.
- **Filter Table selector** - If you want the State Modifier to process only selected alerts, after creating one or more [Filter tables](#), select a Filter Table from the list, to cause state modifier processes to be applied to the selected table. If None is selected, the State Modifier will not apply any filtering to incoming alerts.
- **Edit button** - Causes the Filter Tables table to appear, allowing the creation of one or more Filter tables.

State

Repeat After (sec): 120

Repeat Suffix: Repeat

Escalate After (sec): 300

Escalate Destination: 304

Escalation Suffix: Escalated

Process: White-List Maintenance Edit

The State Modifier uses the following operating rules:

- If a [Filter table](#) is assigned to the State Modifier, incoming alerts are compared to the assigned Filter table using either the Black-List or White-List method, as assigned. Any matching alerts are compared to all of the active alerts previously processed by the State Modifier. If there is not a Filter table assigned to the State Modifier, all incoming alerts are compared to all of the active alerts previously processed by the State Modifier. All alerts, whether processed or not by the State Modifier, pass through to the next downstream block in the system.
- If an alert has an active state and is not currently in the State Modifier active alerts table, the alert will be added to the table
- If an alert has a cancel state and both the message and destination values match an alert in the State Modifier active alerts table, that alert will be removed from the table and the alert with cancel state will be passed downstream in the SNAP processing structure.
- If the age of an alert in the State Modifier active alerts table exceeds any multiple of the Repeat After value, and has not yet exceeded the Escalate After value, a repeat alert will be generated with any non-blank Repeat Suffix added to the original active alert's message
- Once the age of an alert in the State Modifier active alerts table exceeds the Escalate After value, an escalation alert will be generated with any non-blank Escalate Suffix added to the original active alert's message, and will include the original alert's destination value. If the Escalation Destination value is non-blank, an additional escalation alert will be generated with a destination value using the Escalation Destination value. If the Escalate Suffix value is blank, any non-blank Repeat Suffix will be applied to the escalated alert message value.

Monitor Modifier

Monitor None Edit

The Monitor Modifier is designed to handle specialty use cases, such as guard tour, caregiver bed check, and scheduled resident morning check-in.



The primary function of the Monitor Modifier is to monitor for an expected alert message and/or alert destination and to trigger generation of a new alert if the expected alert is not detected as expected. The alert expectation is defined by any combination of an Alert Threshold value and/or an assigned Schedule. The Monitor Modifier has the following elements:

- **Monitor Table selector** - Select application of a Monitor table
- **Monitor Table Editor link** - Create and edit Monitor tables, including a Name value for each table

Close

Monitor Table: My Monitoring BACK

Filter :type any text here + Add New Monitor Record HELP

Enabled	Mon Alert Text	Mon Destination	Schedule	Alert Threshold (min) ▲	New Alert Text	New Destination	Suppress Mon Alerts	Delete
<input checked="" type="checkbox"/>	check-in 202		resident check-in		Failed resident check-in Rm 202	103	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	guard		None	16	Failed guard tour	212	<input checked="" type="checkbox"/>	

Monitor Table records have the following elements:

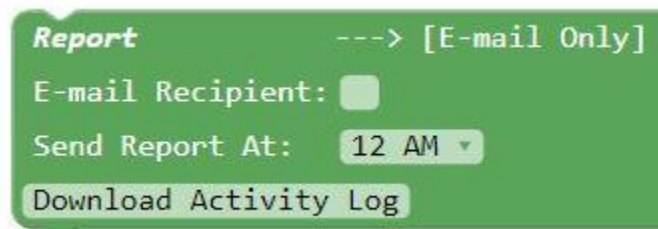
- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Monitor Record** - Prompts for the Name value of a new Monitor Table record
- **Enabled checkbox** - Toggle the application of a Monitor table record
- **Mon Alert Text** - Optionally define the alert text value that will be monitored for
- **Mon Destination** - Optionally define the alert destination value that will be monitored for
- **Schedule** - Optionally assign a schedule for the monitoring process
- **Alert Threshold (min)** - Optionally define a time value, in minutes, that defines how often an alert should be expected to be detected
- **New Alert Text** - Define the alert text value that will be used for the new triggered alert(s)
- **New Destination** - Define the alert destination value that will be used for the new triggered alert(s)
- **Suppress Mon Alerts checkbox** - Toggle the suppression of output of any monitored alerts for the associated Monitor table record
- **Delete** - Delete selected Monitor table record

Following are notes on how the Monitor Modifier operates:

- Incoming alerts are compared to all of the records in the assigned Monitor table
- Monitor table processing will fail to occur for a Monitor table record under any of the following conditions:
 - Both Mon Alert Text value is blank and Mon Destination value is blank
 - Either New Alert Text value is blank or New Destination value is blank. Note that both values must be non-blank.

- If both Alert Threshold value is blank and Schedule value is None
- If a Schedule is assigned and that Schedule is currently inactive
- New alerts can be triggered by any of the following methods:
 - If only Alert Threshold value is assigned, a monitored alert is not detected within the Alert Threshold defined time interval
 - If only a Schedule is assigned, a monitored alert is not detected within any active assigned Schedule interval
 - If both Alert Threshold and Schedule are assigned, a monitored alert is not detected within the Alert Threshold defined time interval during the active intervals of an assigned Schedule
- When both an Alert Threshold and a Schedule are applied to a Monitor table record, the Alert Threshold time interval counter does not start until a particular assigned Schedule interval becomes active. As an example, if a Schedule interval starts at 10:00 AM and the Alert Threshold is set to 5 minutes, and no monitored alerts are detected for that Monitor table record, a new alert will be created at 10:05 AM.
- When only a Schedule is applied to a Monitor table record, if a Monitored Alert is not detected within an active schedule interval, when the active schedule interval ends, a new alert will be immediately created. As an example, if a Schedule interval starts at 10:00 AM and ends at 11:00 PM, a new alert will be created at 11:00 PM.
- The alert message value of all alerts incoming to the Monitor modifier will be examined againsts all records in the assigned Monitor table. If one or more matches occur, the new alert trigger condition requirements will be satisfied and a new alert will not be created. If any of the matching Monitor table records have Suppress Mon Alerts checkbox checked, then the monitored alert will not be passed to the rest of the system.
- Once a new alert is created by the Monitor modifier, that new alert is considered canceled once the monitored alert is detected, at any time, following creation of the new alert, regardless of the state of any assigned Schedule for that Monitor table record. The same detected monitored alert event that can cancel an active Monitor modifier new alert, will also be used to satisfy any defined alert trigger condition requirements.

Report Modifier



The Report Modifier block performs the process of generating daily scheduled reports of alert event activity, when combined with the E-Mail Output Block. The report is output at a selected time, once per day, as an HTML formatted e-mail message to the e-mail recipient defined in the Report Modifier block. The Report Modifier block also supports downloading of a CSV file containing alert history information. The selected report will be compiled and output once per day at the selected time.

Each Report Modifier block compiles its own alert history CSV file, which can be affected by upstream Protocol and/or Modifier blocks, due to their effect on alert object data flow. Each Report Modifier block also monitors all alerts. The Report Modifier retains up to 3 days' worth of activity in the log file.

In order to create an alert history record in the alert history CSV file, an alert needs to meet the following requirements:

- Have an Active state and later have a Cancel state, where both the message and destination values of the alert are the same for both the activation and cancel states. For alerts that are essentially identical other than the destination value, they are individually tracked and create individual alert history records, as required.
- Have a Momentary state

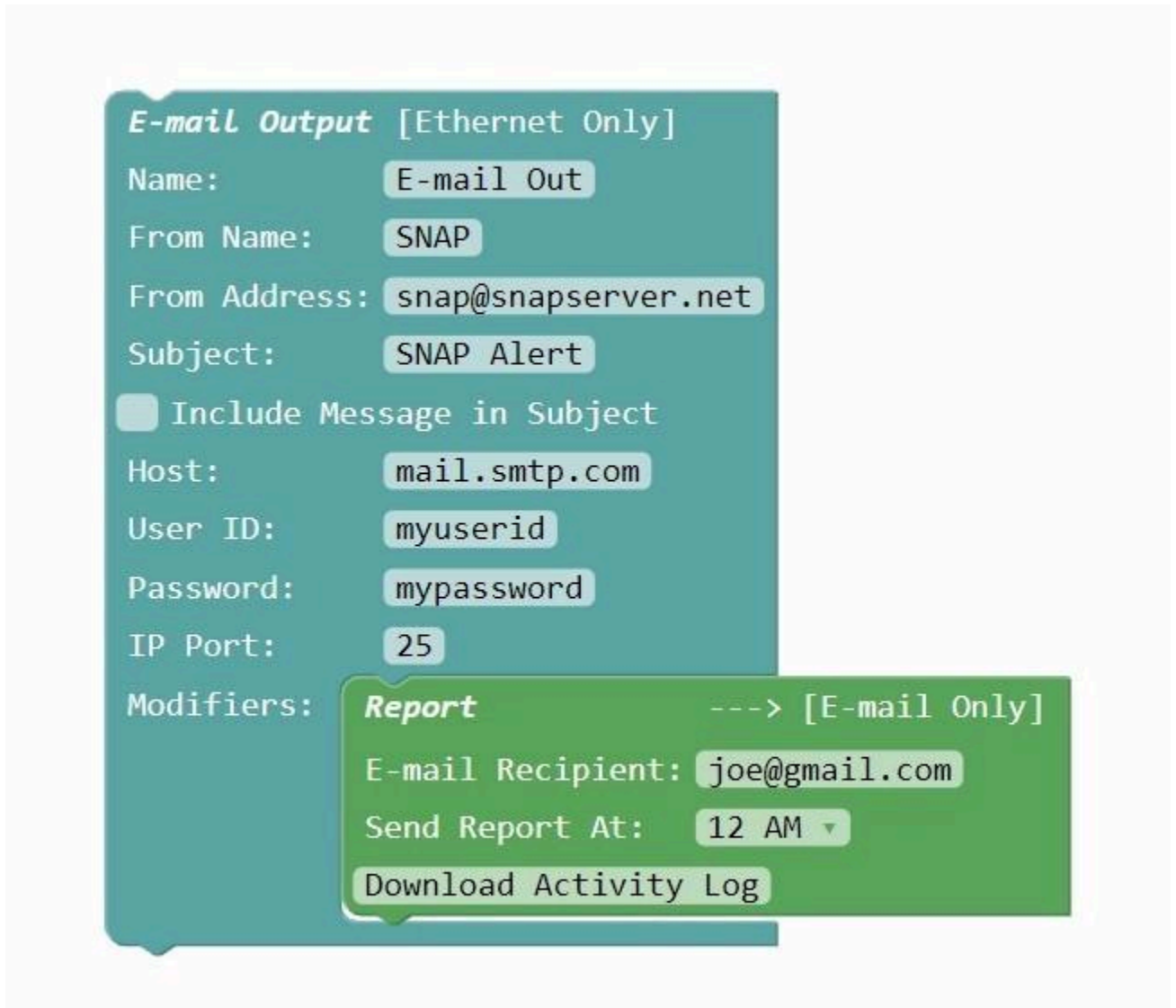
The Report Modifier block compiles alert history information into a CSV file, which is formatted into the following comma delimited fields:

- **message** - The alert message value
- **destination** - The alert destination value
- **activation timestamp** - The alert activation time
- **cancellation timestamp** - The alert cancellation time. If alerts have a Momentary state, they are logged with identical activation and cancellation timestamp values and a duration value of 0
- **duration** - The difference between activation and cancellation timestamps, in units of seconds. If alerts have a Momentary state, they are logged with identical activation and cancellation timestamp values and a duration value of 0.

The Report Modifier Block is composed of the following elements:

- **E-mail Recipient** - The recipient that the selected report will be sent to via the E-mail Output block. If the E-mail Recipient value is invalid or blank, no report will be generated, but the CSV log file will still remain available for download.
- **Send Report At selector** - The hour of the day that the selected report will be sent out via the E-mail Output block. The selected report will be compiled and output once per day at the selected time.
- **Download Activity Log button** - Causes the Filter Tables table to appear, allowing the creation of one or more Filter tables. This file is available for download, regardless of the state of the E-mail Recipient value.

The following image shows a Report Modifier Block inserted into an E-mail Output Block



The following image shows a sample E-mailed Event Activity Report

Event Activity Report

Thursday 05-10-2018 03:41:49 PM

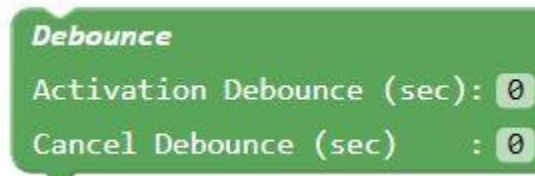
6 total calls, avg response time of 00:13

Start Time	End Time	Alert	Destination	Duration Min:Sec
05-10-2018 03:38:19 PM	05-10-2018 03:38:32 PM	Test message 001	101	00:13
05-10-2018 03:38:24 PM	05-10-2018 03:38:37 PM	Test message 002	101	00:13
05-10-2018 03:38:29 PM	05-10-2018 03:38:42 PM	Test message 003	101	00:13
05-10-2018 03:38:34 PM	05-10-2018 03:38:47 PM	Test message 004	101	00:13
05-10-2018 03:38:39 PM	05-10-2018 03:38:52 PM	Test message 005	101	00:13
05-10-2018 03:38:44 PM	05-10-2018 03:38:57 PM	Test message 006	101	00:13

The following image shows a sample Activity Log CSV file opened in Microsoft Excel

	A	B	C	D	E	F
1	Alert Text	Alert Destination	Alert Start	Alert End	Alert Duration in Seconds	
2	Test message 001	101	5/10/2018 15:38	5/10/2018 15:38	13	
3	Test message 002	101	5/10/2018 15:38	5/10/2018 15:38	13	
4	Test message 003	101	5/10/2018 15:38	5/10/2018 15:38	13	
5	Test message 004	101	5/10/2018 15:38	5/10/2018 15:38	13	
6	Test message 005	101	5/10/2018 15:38	5/10/2018 15:38	13	
7	Test message 006	101	5/10/2018 15:38	5/10/2018 15:38	13	
8	Test message 007	101	5/10/2018 15:38	5/10/2018 15:39	13	
9	Test message 008	101	5/10/2018 15:38	5/10/2018 15:39	13	
10	Test message 009	101	5/10/2018 15:39	5/10/2018 15:39	13	
11	Test message 010	101	5/10/2018 15:39	5/10/2018 15:39	13	

Debounce Modifier



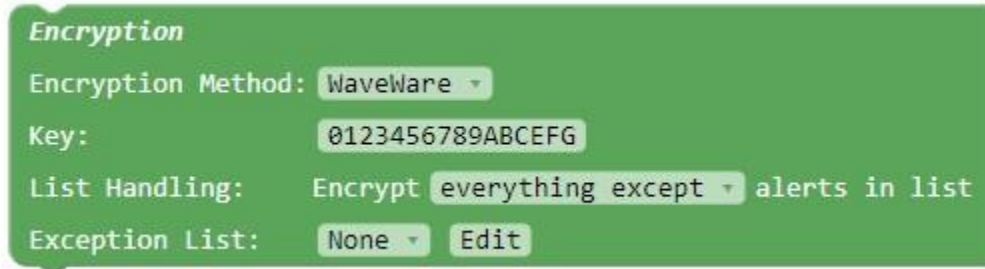
The Debounce Modifier block performs the process of smoothing the notification of alert event activity, for alerts that frequently change state.

The Debounce Modifier Block is composed of the following elements:

- Activation Debounce (sec)** - The quantity of seconds that an alert activation event will be delayed once an alert activation state is initially detected. NOTE: If the affected alert is canceled prior to the Activation Debounce time, the alert activation event will not be generated by the Debounce Modifier block. The alert must remain in an active state for the duration of the Activation Debounce value in order for the Debounce Modifier block to generate an alert activation event. The default value of the Activation Debounce value is zero (0) seconds. A zero (0) value disables the Activation Debounce feature, which allows active alerts to immediately pass into the Debounce Modifier block unaffected.
- Cancel Debounce (sec)** - The quantity of seconds that an alert cancellation event will be delayed once an alert cancellation is initially detected for an active alert. NOTE: If the affected alert becomes active again prior to the Cancel Debounce time, the alert cancellation event will not be generated by the Debounce Modifier block. The alert must remain in a canceled state for the duration of the Cancel Debounce value in order for the Debounce Modifier block to generate an alert cancellation event. The default value of

the Cancel Debounce value is zero (0) seconds. A zero (0) value disables the Cancel Debounce feature, which allows canceled alerts to immediately pass out of the Debounce Modifier block unaffected.

Encryption Modifier



The screenshot shows a configuration panel for the Encryption Modifier block. It has a green header with the word "Encryption" in white. Below the header, there are four rows of configuration options:

- Encryption Method:** A dropdown menu with "WaveWare" selected.
- Key:** A text input field containing the alphanumeric string "0123456789ABCEFG".
- List Handling:** A dropdown menu with "Encrypt everything except" selected, followed by the text "alerts in list".
- Exception List:** A dropdown menu with "None" selected, and an "Edit" button to its right.

The Encryption Modifier block performs the process of tagging alert message values for encryption, in certain SNAP alerts as they pass through the system. The Encryption Modifier can be used multiple times in the system, allowing the configuration of one or more encryption keys for a given site. The Encryption Modifier block allows users to perform simple or complex encrypted messaging use cases, where both encrypted message receiving devices (programmed with either common or individual encryption keys), can be blended with non-encrypted devices into a common SNAP messaging output, and where multiple encryption keys can be deployed. **NOTE:** The Encryption Modifier block should only be applied inside SNAP output protocol blocks, because it modifies (encrypts) the alert message value, and discards alerts which fail to be properly encrypted, which could affect downstream alert processing if used outside of output protocol blocks.

The alerts that are tagged for encryption have their message value encrypted. Currently, the SNAP output blocks that benefit from use of the Encryption Modifier include COMP1, COMP2 and TAP, to allow sending encrypted radio paging messages to WaveWare encrypted pagers.

The Encryption Modifier block, by default, encrypts every alert passing through the block. To selectively encrypt alerts for a given Encryption Modifier block, first configure the **List Handling** method in the Encryption Modifier block to either **Encrypt everything except alerts in list** or to **Encrypt only alerts in list**. Note that all alerts pass through the block, but the combination of selected List Handling method and any assigned Exception List only affects which of the alert message values become encrypted. The goal of selecting a List Handling method is to allow the minimizing of the size of the Exception List table that needs to be configured.

After selecting a List Handling method, select an Exception List (**sec team** in the example). If an Exception List does not exist, click the Edit button in the Encryption Modifier block to create and edit one or more [Exception Lists](#) (which uses a SNAP Filter List table). To edit exception records, select or create a filter table, and ensure that table has records added to it and edited. The example [Exception List table](#) added a record to the **admin encr exceptions** table with a destination value of 103. That means that all incoming alerts will have their Destination (address) value compared to the Destination value of 103, to determine if the assigned encryption rule should be applied to the affected alert.

The Encryption Modifier Block is composed of the following elements:

- **Encryption Method selector - WaveWare** causes alert message values to be encrypted using the WaveWare method, which is compatible with multiple WaveWare pager models that support encrypted messaging.

- **Key** - The encryption key assigned to the Encryption Modifier block. When you select the default WaveWare Encryption Method, you must enter a 16-character Alphanumeric(0-9, A-Z, a-z) string, which should match the decryption string programmed into any WaveWare pagers that will perform decryption. The Key field background will change color when the Key value either contains an illegal character or becomes too long. Key values that are too short will be captured in the SNAP system logs.
- **List Handling selector** - Causes the Individual Encryption Keys table to appear, allowing the creation of one or more Encryption handling records.
- **Exception List selector** - Causes the Individual Encryption Keys table to appear, allowing the creation of one or more Encryption handling records.
- **Edit button** - Causes the Individual Encryption Keys table to appear, allowing the creation of one or more Encryption handling records.

Close

Filter Tables

Filter: type any text here + Add New Filter Table HELP

Name	Edit	Delete
admin encr exceptions	Click Here to Open	
sec team	Click Here to Open	

The Encryption Modifier Filter Tables table allows editing of filter tables that can be assigned to an Encryption Modifier. The Encryption Modifier Filter Tables table has the following elements:

- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Filter Table** - Prompts for the Name value of a new Filter Tables table record. **NOTE:** Filter tables used with the Encryption Modifier block should be named with a reference to encryption, to avoid confusion when using other SNAP modifiers that use Filter tables, as any Filter tables that you create can be applied system-wide.
- **Name** - The value used in the Filter Table Selector of the [Encryption Modifier Block](#)
- **Edit** - Click the link to open the associated Filter Table for editing
- **Delete** - Allows deletion of the selected table record

Close

Filter Table: admin encr exceptions BACK

Filter: type any text here + Add New Filter Record HELP

Key String	Destination	Schedule	Delete
	103	None	

The Filter table allows editing of the selected Filter Table. The Filter table has the following elements:

- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Filter Record** - Creates a blank Filter table record
- **Key String** - Optional key string value that is compared to alert message values in alerts processed by the Filter Modifier
- **Destination** - Optional destination value that is compared to alert destination values in alerts processed by the Filter Modifier
- **Schedule** - Optional schedule assignment to allow processing alerts for this Filter table record only when the assigned [Schedule](#) is active
- **Delete** - Allows deletion of the selected table record

The Encryption Modifier uses the following encryption rules:

- When the Key value is malformed (illegal character, too long, too short, or blank), all alerts that meet the List Handling method and any assigned Exception List will not be processed, will be discarded and logged as an error. Otherwise, all alert message values of alerts that meet the List Handling method and any assigned Exception List will be encrypted using the Encryption Modifier block Key value.
- Incoming alerts are compared to all of the records in the assigned Exception List (Filter Table)
- Destination comparisons to alert message values are case-insensitive
- If **Encrypt only alerts in list** method is assigned in the Encryption block, all alerts that match a record in the assigned Exception List (Filter Table) will be encrypted.
- If **Encrypt everything except alerts in list** method is assigned in the Encryption block, alerts will not be encrypted unless they match one or more records in the assigned Exception List (Filter Table).
- Filter table processing will fail to occur for a Filter table record under any of the following conditions:
 - Both Key String value is blank and Destination value is blank
 - If a Schedule is assigned and that Schedule is currently inactive
- Any non-blank Key String value is compared with an alert message value to see if the Key String is contained within the message, which defines a match between the alert and the Key String
- Any non-blank Destination value is compared with an alert destination value to determine if they match
- When both Key String and Destination values are non-blank in the Filter table record, incoming alerts must match both the Key String and Destination in order for a match to be defined
- When a schedule is assigned to a Filter table record and the schedule is currently inactive, that filter table record will not be processed

Following is an example SNAP system log excerpt that demonstrates how encryption errors are logged when the alert is processed by the associated output blocks that perform alert message encryption controlled by the Encryption Modifier. The log also demonstrates a successful output of an encrypted message. Note that for WaveWare encryption method, a prefix value of **#WW_** is included with all encrypted messages, for purposes of allowing the pager to warn the pager user when invalid encryption related messages are received, so that the invalid encryption related messaging use cases can be quickly resolved by management.

```
2024-03-27 15:37:46> IN: Input Protocol Simulator 'Input Simulator': [101]-"Heartbeat 001"-MOMENTARY
2024-03-27 15:37:46> OUT: 'COMP2 Out' Encoded [101]-"#WW_finAWjceZhYqUR1wNXcyRhBiQg=="-MOMENTARY
2024-03-27 15:37:46> CLEARED BUFFER: '192.168.10.58' Cleared '<LF>Page queued okay<CR><LF><CR><LF>COMP2<GT>'
2024-03-27 15:37:46> OUT: '192.168.10.58' Sent '101<CR><LF>#WW_finAWjceZhYqUR1wNXcyRhBiQg==<CR><LF>'
2024-03-27 15:37:46> IN: '192.168.10.58' Received ''
2024-03-27 15:37:51> System Configuration Changed.
2024-03-27 15:37:56> System Configuration Changed.
2024-03-27 15:38:01> System Configuration Changed.
2024-03-27 15:38:01> IN: Input Protocol Simulator 'Input Simulator': [101]-"Heartbeat 001"-MOMENTARY
2024-03-27 15:38:01> MOD: Error - Encryption Key: 0123456789BCEFG too short, must be 16 char, alert discarded
```

With WaveWare encryption, the following WaveWare pager related use cases can occur:

- **Encrypted message received on unencrypted address/sub address:** The pager will display an alternate warning message of **Encryption not allowed**, along with the address/sub address values that the message was received on. An example address/sub address value is: **1D**, which indicates first address and fourth sub address.
- **Unencrypted message received on encrypted address/sub address:** The pager will display an alternate warning message of **Encryption required**, along with the address/sub address values that the message was received on. The pager will also display a Key icon (🔑) when a message is received on an address/sub address that has encryption enabled. An example address/sub address/key icon value is: **2D🔑**, which indicates second address, fourth sub address, and encryption enabled.
- **Encrypted message received on encrypted address/sub address, but using the wrong Key value:** The pager will display a garbled text message, along with the address/sub address values that the message was received on, and including a key icon.

If you need to use multiple encryption keys per output, use multiple Encryption Modifier blocks, each with a different Key value assigned, and each with a different Exception List assigned. Following is an example.



Group Cancel Modifier



The Group Cancel Modifier block performs the process of generating specialized alert objects for purposes of allowing downstream systems that maintain alert state to be able to auto-remove alerts from their list.

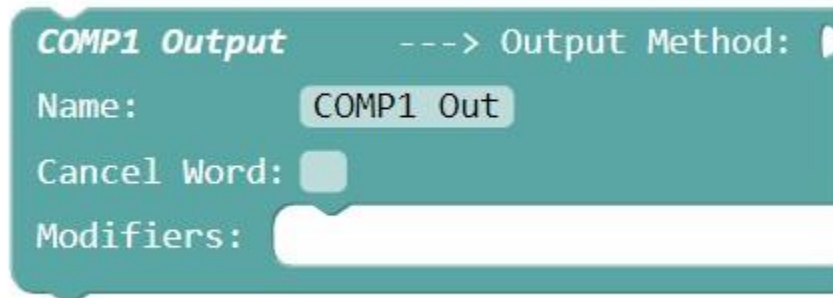
The Group Cancel Modifier Block is required for use only when upstream systems, such as nurse call systems, generate alert cancel events that affect a group of one or more active alerts, and when a SNAP system is configured to use downstream systems that maintain state and automatically remove canceled alerts.

An example of this condition would be a nurse call system feeding a TAP input, which is then routed to a Pager App output, where the nurse call system can generate multiple unique alert activation alert message values, but only generate a common alert cancellation message value to allow the caregiver to know that all active alerts for that group have been canceled.

Note that the creation of these specialized alert cancellation objects will not affect SNAP outputs that do not maintain state, such as the TAP, COMP1 and COMP2 outputs, because they will ignore these alert cancellation objects while processing the original group cancellation alert object as a normal alert output.

Output Protocols

COMP1 Output Protocol



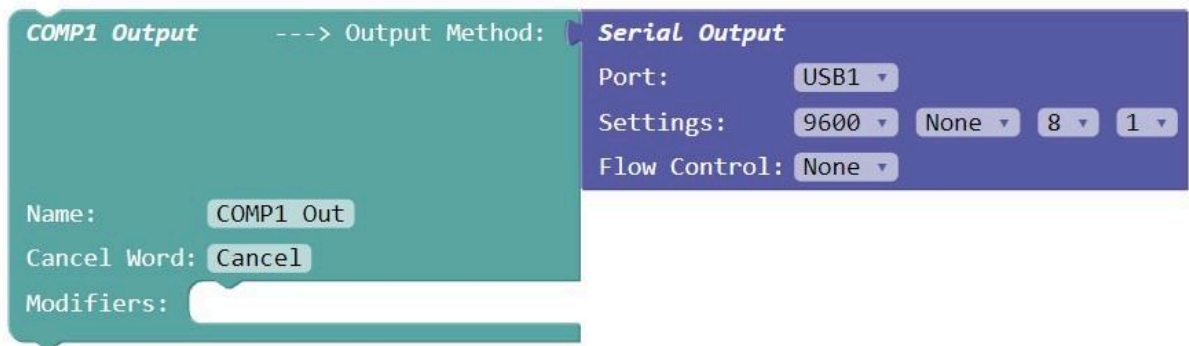
The COMP1 Output block encodes alert data into serial data using the COMP1 protocol, and requires use of one of the SNAP [Output Methods](#). COMP1 output is defined as serial data, with no destination value, delimited by <CR><LF> ASCII control characters, e.g.

This is an alert<CR><LF>

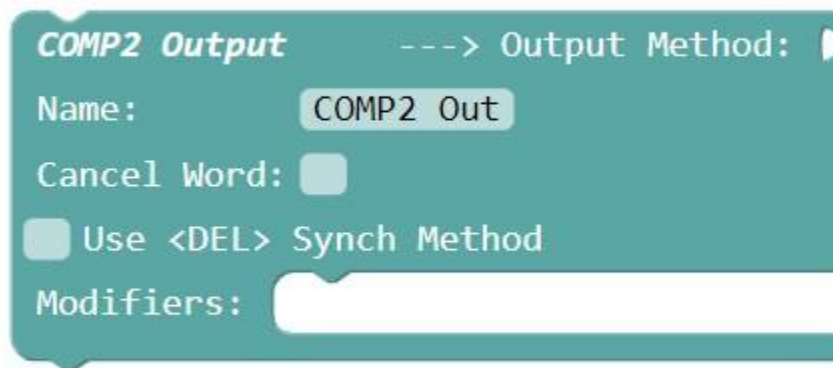
The COMP1 Output block has the following elements:

- **Output Method** - An Output Method is required to be snapped into this block in order for it to function
- **Name** - Used for activity logging
- **Cancel Word** - A case-insensitive word value that, if non-blank, causes the Output Protocol block to generate output data when an alert with a [cancel state](#) is detected. For that data, the Cancel Word value is appended to the output data value.

- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block



COMP2 Output Protocol



The COMP2 Output block encodes alert data into serial data using the COMP2 protocol, and requires use of one of the SNAP [Output Methods](#). COMP2 output is defined as serial data, including both destination and alert message values, delimited by <CR><LF> ASCII control characters, e.g.

102<CR><LF>This is an alert<CR><LF>

The COMP2 Output block has the following elements:

- **[Output Method](#)** - An Output Method is required to be snapped into this block in order for it to function
- **Name** - Used for activity logging
- **Cancel Word** - A case-insensitive word value that, if non-blank, causes the Output Protocol block to generate output data when an alert with a [cancel state](#) is detected. For that data, the Cancel Word value is appended to the output data value.
- **Use Synch Method checkbox** - Allows synchronization of WaveWare paging encoders when using COMP2 protocol
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block

COMP2 Output ---> Output Method: **Ethernet Output**

Name:

Cancel Word:

Use Synch Method

Modifiers:

IP Address:

IP Port:

E-mail Output Protocol

E-mail Output [Ethernet Only]

Name:

From Name:

From Address:

Subject:

Include Message in Subject

Use Authentication

Host:

User ID:

Password:

IP Port:

Modifiers:

The E-mail Output block encodes alert data into e-mail messages using the SMTP protocol. E-mail Output occurs for any incoming alert with an active state and whose destination value is a valid e-mail **To:** address. Alerts passing through the E-mail Output block will have their Destination values compared to the [E-mail Recipients table](#). If an alert's Destination value matches the Destination value in one or more E-mail Recipients table records, then an e-mail will attempt to be delivered to all E-mail Recipients associated with the affected alert. If

there is no match in the E-mail Recipients table, but the alert's Destination value is formatted as an e-mail address, then an e-mail will attempt to be delivered for that alert.

Alerts will pass through the E-mail Output block without modification, to allow downstream blocks to process the same alert objects that entered the E-mail Output block

The E-mail Output block has the following elements:

- **Name** - Used for activity logging
- **From Name** - The optional outgoing e-mail From Name value, e.g. **Sam Jones**
- **From Address** - The outgoing e-mail From Address value, e.g. **sam.jones@test.net**
- **Subject** - The optional outgoing e-mail Subject value
- **Include Message in Subject checkbox** - If checked, the alert message value is used as the outgoing e-mail subject line
- **Use Authentication checkbox** - If checked, the User ID and Password values will be used to perform authorization with the SMTP server. If unchecked, message delivery will be attempted without using authorization credentials.
- **Host** - The outgoing mail server URL or IP address
- **User ID** - The User ID value required by the outgoing mail server
- **Password** - The Password value required by the outgoing mail server
- **IP Port** - The IP Port value that is used to connect to the outgoing mail server
- **Recipients button** - Clicking this button opens the [E-mail Recipients table](#)
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block

E-mail Output [Ethernet Only]

Name: E-mail Out

From Name: SNAP

From Address: snap@wirelessmessaging.com

Subject: SNAP Alert

Include Message in Subject

Use Authentication

Host: smtpout.secureserver.net

User ID: snap@wirelessmessaging.com

Password: mySNAPPW

IP Port: 25


Recipients

Modifiers:

Close

E-mail Recipients

Filter :type any text here + Add New E-mail Recipient HELP

Destination	E-mail Address	Delete
101	snap@wirelessmessaging.com	

The E-mail Recipients table causes the [E-mail](#) Output block to translate alert Destination values to E-mail addresses, as required.

If an alert Destination value matches the Destination value in one or more table rows, all affected table rows will be processed.

The E-mail Recipients table has the following elements:

- **Destination** - The value that alert destination values will be compared to. If there is a match between this Destination value and an alert destination value, then an e-mail will attempt to be delivered that uses the associated E-mail Address value for the affected record(s).

- **E-mail Address** - The To: address value that will be used for attempted e-mail delivery when a Destination match occurs
- **Delete** - Allows deletion of the selected table record

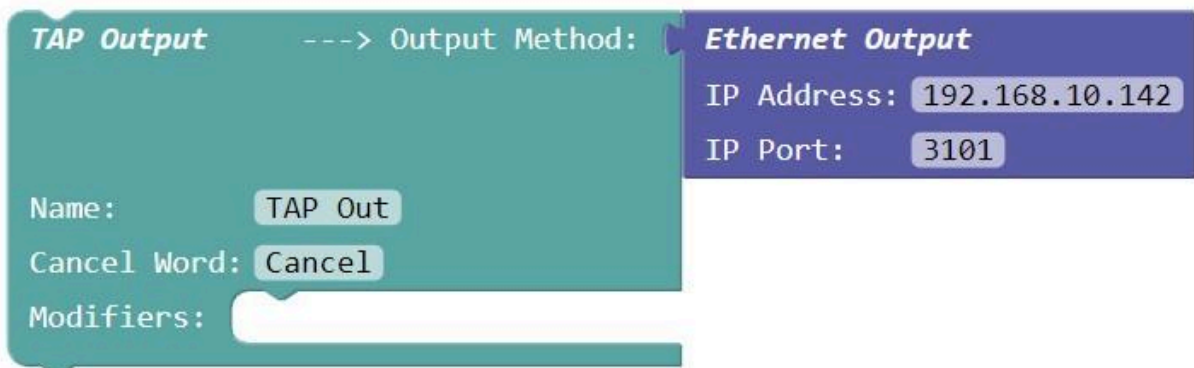
TAP Output Protocol



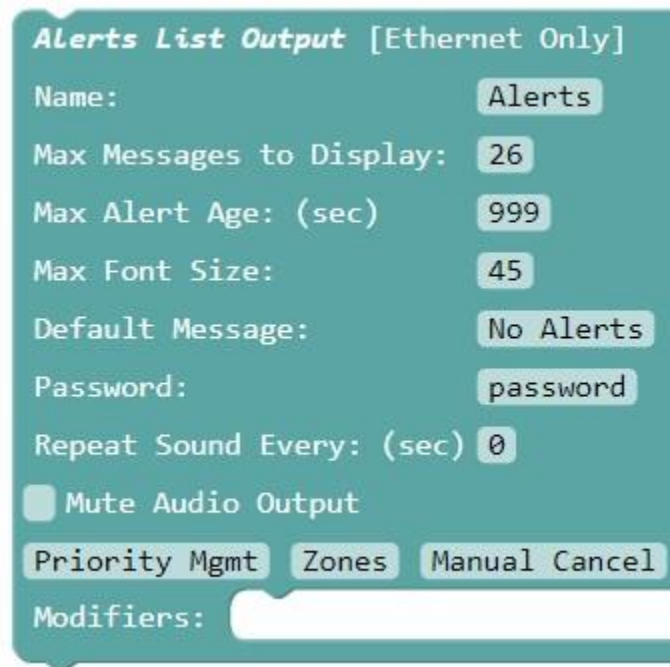
The TAP Output block encodes alert data into serial data using the TAP protocol, and requires use of one of the SNAP [Output Methods](#). TAP output is defined as serial data, including both destination and alert values, delimited by ASCII control characters, and including a checksum. TAP is similar to COMP2, but more complex with regard to bi-directional communications between the server and client.

The TAP Output block has the following elements:

- [Output Method](#) - An Output Method is required to be snapped into this block in order for it to function
- **Name** - Used for activity logging
- **Cancel Word** - A case-insensitive word value that, if non-blank, causes the Output Protocol block to generate output data when an alert with a [cancel state](#) is detected. For that data, the Cancel Word value is appended to the output data value.
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block



Alerts List Output Protocol



The image shows a configuration window titled "Alerts List Output [Ethernet Only]". It contains several input fields and buttons. The fields are: "Name:" with value "Alerts", "Max Messages to Display:" with value "26", "Max Alert Age: (sec)" with value "999", "Max Font Size:" with value "45", "Default Message:" with value "No Alerts", "Password:" with value "password", and "Repeat Sound Every: (sec)" with value "0". There is a checkbox for "Mute Audio Output" which is currently unchecked. Below these fields are three buttons: "Priority Mgmt", "Zones", and "Manual Cancel". At the bottom, there is a "Modifiers:" label followed by a text input field.

The Alerts List Output block encodes alert data into a dynamic list of active alerts, which is output to one or more browsers. The Alerts List is [presented with a black background and white text](#). Each alert item includes a circular icon indicating the priority of the alert. A white colored circular icon indicates normal priority while other colors signify higher priority alerts. The outside border of the Alerts List indicates the color of the highest priority alert currently being displayed in the list. Black is the default border color, indicating that there are currently no higher priority alerts being displayed.

The prioritization of alerts in the Alerts List output is based upon the configuration of the [Alerts List Priority Management](#) table.

To view the Alerts List in a browser, log into the `/alerts` sub-URL of SNAP, e.g. `192.168.10.142/alerts`, using the `alerts` username value and the Password value defined in the Alerts List Output block. Alternatively, you can log into the SNAP primary URL with a username value of `alerts`, to cause the Alerts List to appear.

Messages are automatically added to and removed from the list displayed on the Alerts List, based on the Alerts List Output block settings and on the optional destination parameter in the Alerts List connection URL.

You can configure desktop shortcut URLs to do the following with the Alerts List Output:

- Automatically connect and Log into the Alerts List page of the SNAP server using the required login credentials
- Modify the Alerts List Name value displayed at the top of the Alerts List, using the optional name parameter in the URL. If you omit the name parameter in the URL, the Alerts List Name value will be obtained from the Name value defined in the Alerts List Output block.
- Cause the Alerts List output to filter alerts by their Destination value, using the optional destination parameter in the URL. If you omit the destination parameter in the URL, the Alerts List will display all of the alerts going through the Alerts List Output block in the SNAP system.

An example URL for doing all of the above is:

192.168.10.241/alerts?name=East%20Wing&destination=312&username=alerts&password=ALPass

Note that the SNAP destination values can be alphanumeric strings. Another example URL using an alphanumeric destination value of **west wing** is:

192.168.10.241/alerts?name=Wast%20Wing&destination=west%20wing&username=alerts&password=password

Following is another example URL telling the system to use the Name value defined in the Alerts List Output block and to use no destination filtering:

192.168.10.312/alerts?username=alerts&password=ALPass

The Alerts List Output block has the following elements:

- **Name** - The reference Name that will display at the top of the Alerts List screen, and will be used for activity logging
- **Max Messages to Display** - Messages are displayed on the Alerts List in a FIFO (first-in first-out) order, with oldest messages replaced by newest as the Max Messages value is reached in the size of the displayed list
- **Max Alert Age (sec)** - An optional maximum age an active alert will be allowed to be displayed in the Alerts List. When the state of an alert becomes **cancel**, the alert will be automatically removed from the list. If the Max Alert Age value is blank or 0, messages will remain on the list until the associated alert goes into a Cancel state. Alerts with a Momentary state remain on the list until the Max Alert Age value occurs for those alerts. Allowable Max Alert Age values include blank, and 0 through 9999.
- **Max Font Size** - The Alerts List font size is automatically adjusted to accommodate the entire displayed active list. As the size of the list decreases, the font size automatically adjusts larger, until it reaches the Max Font Size value. Alternately, as the size of the alert list grows, the font size is automatically adjusted downward. Allowable Max Font Size values range from 20 to 300.
- **Default Message** - An optional message that appears on the Alerts List when there are no currently active alerts to be displayed.
- **Password** - The site wide password required when logging into SNAP using the **alerts** username value
- **Repeat Sound Every (sec)** - An optional audio output sound repeating feature that affects Alert List operation in both browser and in the DAV. Default operation is that the audio alert output will occur once on each activation of a new alert. When this value is greater than 0, the audio alert will be repeated while one or more alerts are active in the Alerts List Output. If the Repeat Sound Every value is blank or 0, audio output will only occur once per each new alert. Allowable Repeat Sound Every values include blank, and 0 through 9999.
- **Mute Audio Output checkbox** - When checked, audio sound will not be played on alert update events on the Alerts List outputs
- **Priority Mgmt button** - Clicking this button opens the [Alerts List Priority Management table](#)
- **Zones button** - Note: The Zones button is only visible if your system has the Alerts List Zones feature enabled. Clicking this button opens the [Alerts List Zones table](#)
- **Manual Cancel button** - Note: The Manual Cancel button is only visible if your system has the Alerts List Zones feature enabled. Clicking this button opens the [Manual Cancel - Alerts List Output table](#)

- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block

Alerts List Output [Ethernet Only]

Name: SNAP Alerts

Max Messages to Display: 10

Max Alert Age: (sec) 250

Max Font Size: 45

Default Message: Smile!

Password: ALPass

Mute Audio Output




Priority Mgmt Zones Manual Cancel

Modifiers:

Close

Alerts List Priority Management

Filter :type any text here + Add New Priority Mgmt Rule HELP

Key String	Destination	Color	Priority Level	Priority Sound	Delete
code red		Red	Priority 1	<input checked="" type="checkbox"/>	
code green		Green	Priority 2	<input checked="" type="checkbox"/>	
code blue		Blue	Priority 3	<input checked="" type="checkbox"/>	

The Alerts List Priority Management table causes the [Alerts List](#) Output block to apply priority and color enhancements to displayed alerts, if Key String and/or Destination matches occur, for a particular alert. The color enhancements affect the color of the bullet icon that is next to an affected alert.

If Key String and Destination fields are both blank, that table row will not be processed. If both Key String and Destination fields are non-blank, an alert must match both Key String and Destination in order for the color and priority enhancements to be applied. If an alert matches a particular table row, no more table rows will be processed for the affected alert.

The Alerts List Priority Management table has the following elements:

- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Priority Mgmt Rule button** - Clicking this button prompts for a Key String value and creates a new table row
- **Key String** - String value that is compared to alert message values in alerts processed by the Alerts List Output block. When an alert message value contains a Key String value, the associated Priority, Color,

and Priority Sound assignments will be applied to the affected alert as it is displayed on all affected Alert Lists.

- **Destination** - The value that alert destination values will be compared to. If there is a match between this Destination value and an alert destination value, and the Key String value does not cause a conflict, then the associated Priority, Color, and Priority Sound assignments will be applied to the affected alert as it is displayed on all affected Alert Lists.
- **Color** - The color that will be applied to the bullet displayed next to a matching alert in the Alerts List output. If the matching alert is the highest priority alert in the list, that color assignment will also be applied to the outside border of the Alerts List display screen..
- **Priority Level** - The Priority Level that should be applied to a matching alert.
- **Priority Sound checkbox** - If checked, when the matching alert first appears on the Alerts List display, the priority sound will be output, as long as Mute Audio Output is not checked in the Alerts List Output block.
- **Delete** - Allows deletion of the selected table record



Username	Zone Name	Destination	Delete
eastwing	East Wing	101	
eastwing		102	
westwing	West Wing	103	

The Alerts List Zones table causes the [Alerts List](#) Output block to change the title text on any Alerts List display that logs in with a Username value defined in the Alerts List Zones table. In addition, the Alerts List Zones table can be used to filter the affected Alerts List display based upon any Destination value defined for a given Username.

The Username field is mandatory. The Zone Name field and the Destination field are optional. If you want to filter the Alerts List display by Destination, you need to create one record per each Destination value that you want to have displayed on the affected Alerts List display.

For an Alerts List display that has logged in with a given Username that is in the Alerts List Zones table, if a Zone Name field value is non-blank for any of the affected records, the Title at the top of the Alerts List display will be changed to reflect the Zone Name.

For a given record, if the Zone Name and Destination fields are both blank, that table row will not be processed.

For an Alerts List display that has logged in with a given Username that is in the Alerts List Zones table, each associated Alerts List Zones record that has a non-blank Destination value will cause only alerts to be filtered out unless their Destination value matches any of the associated Destination values in the Alerts List Zones table.

The Alerts List Zones table has the following elements:

- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Alerts List Zoning Rule button** - Clicking this button creates a new blank table row
- **Username** - String value that is compared to the username of logged in Alerts List display units, to determine if the display title needs to be changed and/or the alerts filtered by destination.
- **Zone Name** - String value that is used to change the display title of Alerts List display units logged in with the associated Username.
- **Destination** - The value that alert destination values will be compared to. If there is a match between this Destination value and an alert destination value, for an Alerts List display unit logged in with the associated Username, then the alert will be displayed on the affected display unit.
- **Delete** - Allows deletion of the selected table record

Manual Cancel - Alerts List Output

Destination	Message	Cancel
101	Test message 001	X
101	Test message 002	X
101	Test message 003	X
101	Test message 004	X
101	Test message 005	X
101	Test message 006	X
101	Test message 007	X
101	Test message 008	X

The Manual Cancel - Alerts List Output table allows administrators or system level users to be able to manually cancel active alerts from a web page.

The Manual Cancel - Alerts List Output table is dynamically updated by system level alert creation/cancellation events.

Following is an example image of an Alerts List output, demonstrating the following:

- The [Alerts List Priority Management](#) table has been configured to apply blue and green colors as well as priority levels to affected alerts. In this example, the blue color is associated with a higher priority level than the green color, causing the border of the Alerts List display to appear blue. Note that two of the alerts in the list are identified as higher priority alerts, due to the non-white color of the circular icons, and the presence of a non-black screen border.
- The Alerts List name value is the default value of **SNAP Alerts** but can be configured to display with a custom reference value such as **2 East**, using either the Name value defined in the Alerts List Output block or the Zone Name value defined in the [Alerts List Zones table](#).
- The age of each alert is represented to the right of each alert, with the newest alert appearing at the bottom of the list

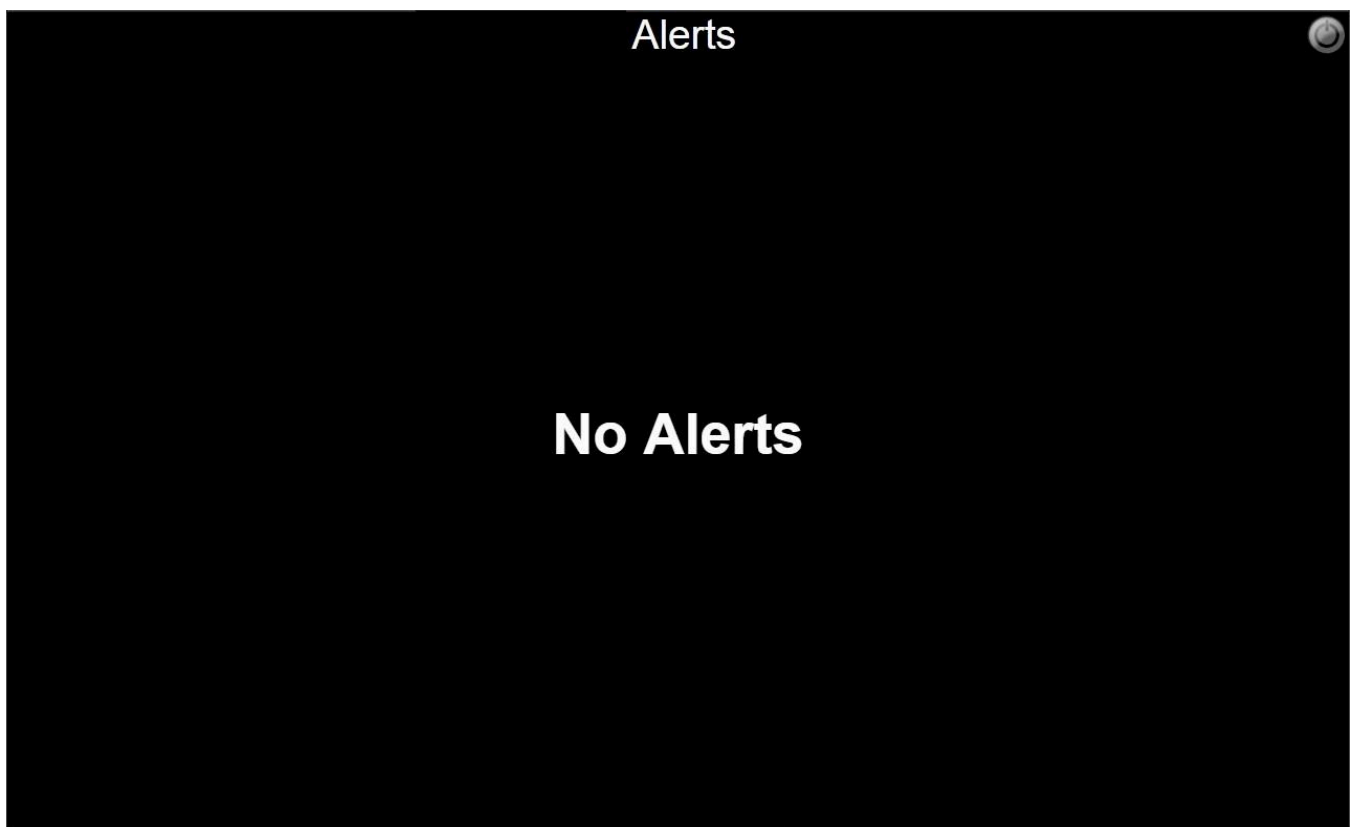
- The Font size of the Alerts List text is auto-adjusted to be able to display the entire alerts list in the screen, with the font size increasing as the quantity of alert messages decrease, up to the Max Font Size value defined in the Alerts List Output block.
- The logout icon appears in the top right of the display, to allow you to log out of the Alerts List display at any time



Following is an example image of an Alerts List output where the higher priority blue colored alert has been removed and the next highest priority active alert causes the screen border to turn green



Following is an example image of an Alerts List default output



Pager App Output Protocol

Pager App Output [Ethernet Only]

Name: Pager App Out

Site Password: password

Priority Mgmt

Modifiers:

The Pager App Output block encodes alert data into serial data and requires use of a [Pager App](#). The Pager App can display normal and priority messages, with active alerts being able to be auto-acknowledged by the server or manually acknowledged by the app user via long touch on an active alert list item in the app.

Note that the quantity of concurrent Pager App connections is limited by licensing, and that additional concurrent Pager App connections capacity can be obtained by purchasing additional licenses.

The Pager App Output block has the following elements:

- **Name** - Used for activity logging
- **Site Password** - The password that any Pager App must use in order to successfully connect to the SNAP Pager App output.
- **Priority Mgmt button** - Clicking this button opens the [Pager App Priority Management table](#)
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block

[Close](#)

Pager App Priority Management

Filter :type any text here + Add New Priority Mgmt Rule [HELP](#)

Key String	Destination	Color	Priority Level	Delete
red		Red	Priority 2	
green		Green	Priority 1	
yellow		Yellow	Priority 1	
blue		Blue	Priority 3	

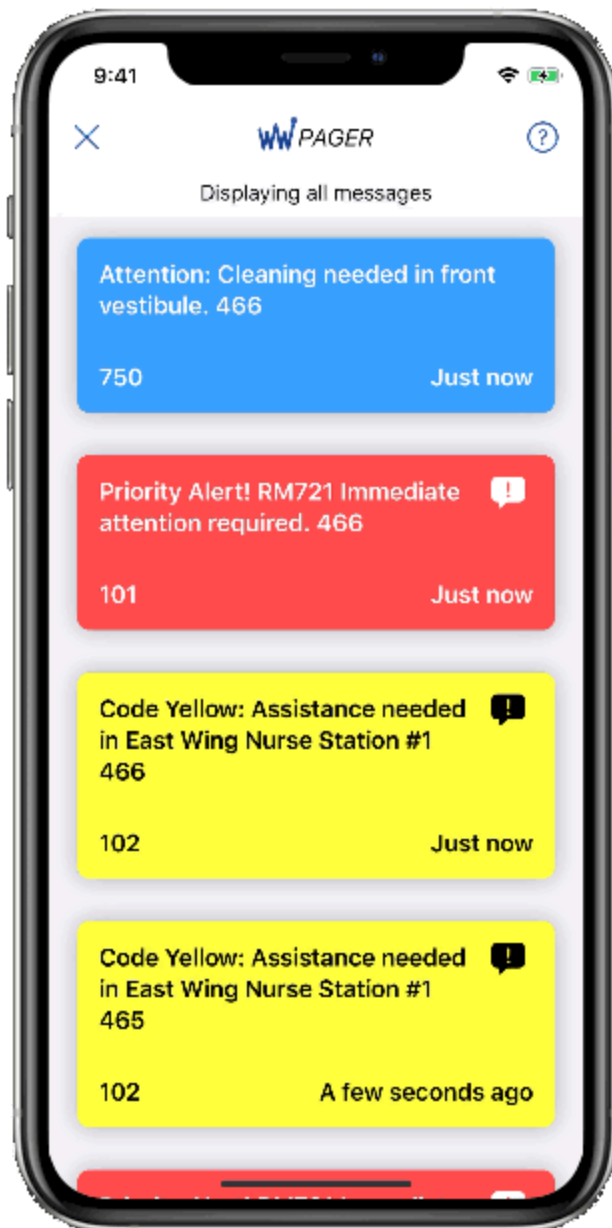
The Pager App Priority Management table causes the [Pager App](#) Output block to apply priority enhancements to displayed alerts, if Key String and/or Destination matches occur, for a particular alert

If Key String and Destination fields are both blank, that table row will not be processed. If both Key String and Destination fields are non-blank, an alert must match both Key String and Destination in order for the priority enhancements to be applied. If an alert matches a particular table row, no more table rows will be processed for the affected alert.

The Pager App Priority Management table has the following elements:

- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Priority Mgmt Rule button** - Clicking this button prompts for a Key String value and creates a new table row
- **Key String** - String value that is compared to alert message values in alerts processed by the Pager App Output block. When an alert message value contains a Key String value, the affected alert will be marked as a priority alert.
- **Destination** - The value that alert destination values will be compared to. If there is a match between this Destination value and an alert destination value, and the Key String value does not cause a conflict, then the affected alert will be marked as a priority alert.
- **Color** - The color that will be assigned to a matching alert when displayed in the app. A default color will be applied to all alerts that do not match a record in this table.
- **Priority Level** - The priority level that will be assigned to a matching alert when displayed in the app. For alerts not matching a record in this table, no priority will be assigned.
- **Delete** - Allows deletion of the selected table record

Following is an example image of the Pager App



9:41

WV PAGER

Displaying all messages

Attention: Cleaning needed in front vestibule. 466

750

Just now

Priority Alert! RM721 Immediate attention required. 466

101

Just now

Code Yellow: Assistance needed in East Wing Nurse Station #1 466

102

Just now

Code Yellow: Assistance needed in East Wing Nurse Station #1 465

102

A few seconds ago

CCTV Text Overlay Output Protocol

CCTV Text Overlay Output ---> [Ethernet Only]

Name:

Header:

Footer:

Cancel Word:

IP Port:

Modifiers:

The CCTV Text Overlay Output block encodes alert data into serial data that is compatible with POS input interfaces on CCTV systems. CCTV Text Overlay output is defined as serial data, including an optional Header value, the Name value with auto timestamping, SNAP system identifier, an alert message value, and an optional Footer value, delimited by <CR><LF> ASCII control characters, e.g.

Header Value

CCTV Text Overlay Out 04-10-18 03:50:42PM

SNAP Ver 1.01 S/N 123459876

This is an alert

Footer Value

The CCTV Text Overlay Output block has the following elements:

- **Name** - Included in the second CCTV output field, accompanied by the timestamp value. Also Used for activity logging
- **Header** - An optional Header value that can benefit the ability to parse the CCTV Text Overlay output in the attached CCTV system
- **Footer** - An optional Footer value that can benefit the ability to parse the CCTV Text Overlay output in the attached CCTV system
- **Cancel Word** - A case-insensitive word value that, if non-blank, causes the Output Protocol block to generate output data when an alert with a [cancel state](#) is detected. For that data, the Cancel Word value is appended to the output data value.
- **IP Port** - The IP Port value used for CCTV systems to connect to the SNAP CCTV Text Overlay Output server
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block

CCTV Text Overlay Output ---> [Ethernet Only]

Name: Hal's Security

Header: ++++++

Footer: *****

Cancel word:

IP Port: 10001

Modifiers:

HTTP Output Protocol

HTTP Output [Ethernet Only]

Name: HTTP Out

URL:

Auth Method: None ▾

Username:

Password:

Retries: 2

Retry Delay (sec): 1

Output Cancel Events

Modifiers:

The HTTP Output block encodes alert data into serial data that is output to a remote HTTP server using the GET method.

The HTTP Output block provides configuration of the HTTP Output and includes the following elements:

- **Name** - Used for activity logging
- **URL** - The URL used for connecting to the remote HTTP server. The URL must include one or more alert parameter placeholders in it, including:
 - **{Destination}** - A placeholder for the Destination value of the alert being output. It is case-insensitive, but must be spelled out, including the curly braces. Destination values will be URI encoded before being inserted into the final URL value.
 - **{Message}** - A placeholder for the Message value of the alert being output. It is case-insensitive, but must be spelled out, including the curly braces. Message values will be URI encoded before being inserted into the final URL value.

- **{State}** - A placeholder for the State value of the alert being output. It is case-insensitive, but must be spelled out, including the curly braces. Possible state values include active, cancel, and momentary
- Examples of properly formatted URL values include:
 - **http://192.168.10.56/api?destination={Destination}&message={Message}&state={State}**
 - **http://192.168.10.56/api?message={message}**
 - **http://12.63.10.47?target={destination}&msg={message}**
- **Auth Method** - The method of authentication required for connection to the remote HTTP server, with values of None, Basic or Digest. If None is selected, Username and Password values will not be used.
- **Username** - An optional value included for logging into the remote HTTP server, when Basic or Digest authentication method is selected
- **Password** - An optional value included for logging into the remote HTTP server, when Basic or Digest authentication method is selected
- **Retries** - The quantity of attempts to be made to deliver an alert to the remote HTTP server before an error is logged
- **Retry Delay (sec)** - The amount of time, in seconds, between retry attempts to be made to deliver an alert to the remote HTTP server
- **Output Cancel Events checkbox** - When checked, alert cancel event notification will be output to the remote HTTP server, in addition to the normal active and momentary state values
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block

HTTP Output [Ethernet Only]

Name:

URL:

Auth Method:

Username:

Password:

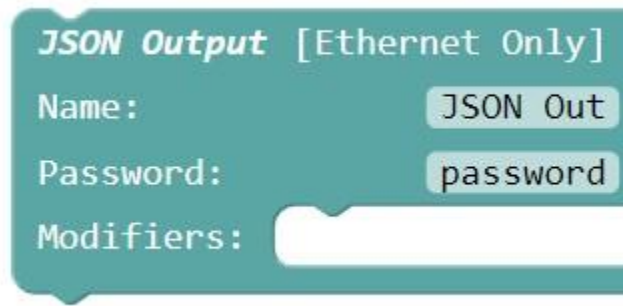
Retries:

Retry Delay (sec):

Output Cancel Events

Modifiers:

JSON Output Protocol

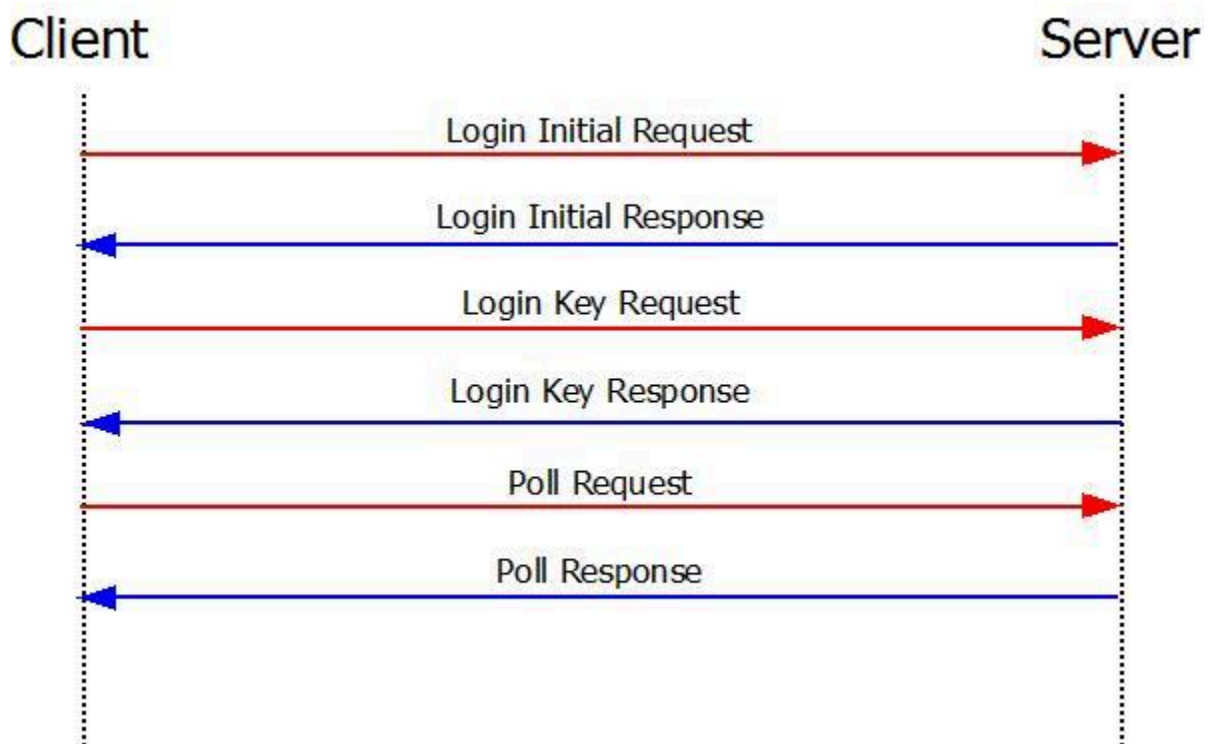


The JSON Output block encodes alert data into serial data that is output in response to HTTP connections at the SNAP system sub URL value of `/json`, e.g. `192.168.10.142/json`

The JSON Output block provides configuration of the JSON Output and includes the following elements:

- **Name** - Used for activity logging
- **Password** - The value required to be included with a md5 hashed key parameter in the HTTP GET or POST data when a device connects to the SNAP JSON Output server. If the key value is incorrect, the JSON output will not occur.
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block

Following is documentation on the SNAP JSON Output protocol, where the SNAP JSON Output acts as the Server:



- **Login Initial Request** - Client connects to JSON Output server URL, e.g. **192.168.10.142/json**
- **Login Initial Response** - Server responds with a session value and the current server timestamp, e.g. a current Unix time value of 1524158994
- **Login Key Request** - Client connects with an md5 hashed value that combines the received timestamp value and the JSON Output Password value, e.g. a timestamp value of 1524158994 and a Password value of password are concatenated into a string value of **1524158994password** and md5 hashed into a key value of **c9e930a16743a339fb552daa950f6eb7** to create a connection URL value of, for example, **http://127.0.0.1/json?session=FVX7W2X061SAEZ9P&key=c9e930a16743a339fb552daa950f6eb7**. The Password value, that the server expects to be extracted from the key value, is defined in the JSON Output block configuration.
- **Login Key Response** - If the Login Key Request is successful, the server will respond with a JSON list of all active alerts, including momentary alerts that occurred within the previous 60 seconds of the response. The response will also include a session parameter. If the Login Key Request is unsuccessful, the server will respond with a JSON formatted error message.
- **Poll Request** - Client connects with a URL value that includes the session and, if not blank, the last message id received, to allow the server to synchronize which messages to respond to the Poll Request with for the given session, such as **http://127.0.0.1/json?session=FVX7W2X061SAEZ9P&last_id=2**
- **Poll Response** - If the session is valid, and 60 seconds have not elapsed since the connection event, the server will respond with an alert. If 60 seconds have elapsed since the connection event, the server responds with a blank alert. If the session is invalid due to either a session timeout or other reason, the server will respond with a JSON formatted error message. If an error occurs, the client should attempt a Login Initial Request.

The JSON Poll Response elements for an active alert notification include:

- **timestamp** - The quantity of seconds since the alert was detected by the server
- **message** - The alert message value
- **destination** - The alert destination value
- **state** - The alert [state](#) value
- **id** - The alert id value, used by the JSON client to pass the last_id parameter in the next poll request
- **session** - The session value defined by the server during the login process

A typical JSON Output Poll Response might look like the following:

```
{'alarms': [{'timestamp': 46, 'message': 'this is a test 1', 'destination': '101', 'state': 'active', 'id': 34}], 'session': 'I1CLIACI5XP2RMU2'}
```

or when there are no alerts to report:

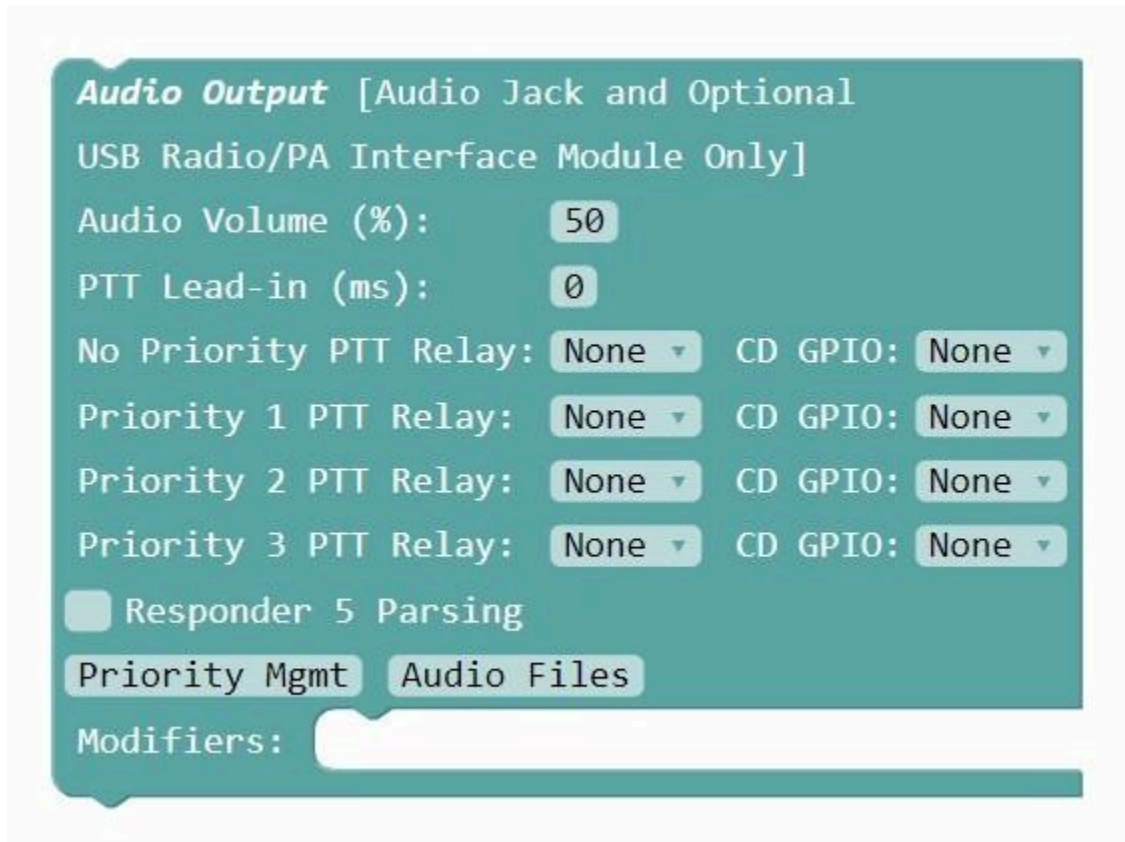
```
{'alarms': [], 'session': 'I1CLIACI5XP2RMU2'}
```

Following are JSON Output error response examples:

- **{'error': 'Protocol Not Enabled'}** - JSON Output block is not active in the SNAP configuration.
- **{'error': 'No Valid Salt String for Session'}** - The Login Key Request contained an invalid salt string value, requiring client to attempt Login Initial Request

- **{'error': 'Authentication Failed'}** - Likely due to the Password value configured in the JSON Output block not matching the Password value hashed into the Login Key Request, requiring a new Password value to be configured into the JSON client
- **{'error': 'Session ID Invalid'}** - Wrong session value provided, likely due to session timeout in server, requiring client to attempt Login Initial Request

Audio Output Protocol



The Audio Output block encodes alert data into an audio stream suitable for output over radio systems, PA systems, and/or speaker systems. Audio is output from an optional Peavey USB-P audio interface adapter, or from the headphone jack, if the Peavey USB-P audio interface adapter is not attached. If an optional [USB Radio/PA Interface Module](#) is attached to SNAP, the Audio Output block will also interact with that module as a means of performing Carrier Detect and PTT signaling with a radio or PA system.

The Audio Output block provides configuration of audio output and includes the following elements:

- **Audio Volume (%)** - The audio volume level that will be output via the headphone jack on the SNAP system, in units of percentage. Range is 10 to 100
- **PTT Lead-in (ms)** - The amount of time, in milliseconds, that the PTT relay will be activated before audio is output. Range is from 0 to 1000.
- **No Priority PTT Relay selector** - Allows selection of None or PTT relay 1 through 4 for the lowest priority level (Normal alerts)
- **No Priority CD GPIO selector** - Allows selection of None or GPIO pin 1 through 4 for carrier detection signal monitoring for the lowest priority level (Normal alerts)
- **Priority 1 PTT Relay selector** - Allows selection of None or PTT relay 1 through 4 for priority level 1

- **Priority 1 CD GPIO selector** - Allows selection of None or GPIO pin 1 through 4 for carrier detection signal monitoring for the priority level 1
- **Priority 2 PTT Relay selector** - Allows selection of None or PTT relay 1 through 4 for priority level 2
- **Priority 2 CD GPIO selector** - Allows selection of None or GPIO pin 1 through 4 for carrier detection signal monitoring for the priority level 2
- **Priority 3 PTT Relay selector** - Allows selection of None or PTT relay 1 through 4 for priority level 3
- **Priority 3 CD GPIO selector** - Allows selection of None or GPIO pin 1 through 4 for carrier detection signal monitoring for the priority level 3
- **Responder 5 Parsing** - If checked, the Audio Output block will parse the alert message value assuming it is formatted using a specially configured Rauland Responder format that includes either asterisk (*) or colon (:) delimiters, as a means of identifying area, room, bed and status parameter values from the alert message.
- **Audio Files button** - Allows uploading of audio files to be used by the Audio Output block
- **Priority Mgmt button** - Clicking this button opens the [Audio Priority Management table](#)
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block

Audio Output [Audio Jack and Optional USB Radio/PA Interface Module Only]

Audio Volume (%): 50

PTT Lead-in (ms): 100

No Priority PTT Relay: 1 ▾ CD GPIO: 1 ▾

Priority 1 PTT Relay: 2 ▾ CD GPIO: 2 ▾

Priority 2 PTT Relay: 3 ▾ CD GPIO: None ▾

Priority 3 PTT Relay: 3 ▾ CD GPIO: None ▾

Responder 5 Parsing

Priority Mgmt Audio Files

Modifiers:

[Close](#)

Audio Output Priority Management

Filter :type any text here

[+ Add New Priority Mgmt Rule](#) [HELP](#)

Key String	GPIO Pin	Audio File	Priority Level	Repeats	Delete
patient	None	patient.mp3	Priority 1	2	
	4	alarm.mp3	None	0	

The Audio Priority Management table causes the [Audio](#) Output block to be able to use key string detection as a method of deciding which audio file to output, which priority level to apply, if any, and the quantity of audio stream repeats to perform.

The Audio Priority Management table has the following elements:

- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Priority Mgmt Rule button** - Clicking this button creates a new table row
- **Key String** - An optional string value that is compared to alert message values in alerts processed by the Audio Output block. When an alert message value contains a Key String value, the associated Audio File will be output, using the associated Priority Level, and repeated as defined by the Repeats value.
- **GPIO Pin** - An optional drop down selection that, when selected, causes the Audio Output block to monitor the attached USB Radio/PA Interface Module's associated GPIO pin for a pulled-to-zero contact closure event. When the associated GPIO Pin has a pulled-to-zero contact closure event, the associated Audio File will be output, using the associated Priority Level, and repeated as defined by the Repeats value.

- **Audio File** - The audio file to be output representing the main body of the alert message. If Responder 5 Parsing is enabled, audio files associated with area, room, and bed may also be output. If Responder 5 Parsing is disabled, only this associated audio file will be output on a key string match.
- **Priority Level** - The priority level that will be applied to the associated alert, with None being the normal processing level. The priority level values range from 1 to 3, with 3 being the top priority for output processing.
- **Repeats** - The quantity of times that the associated audio stream will be repeated after the initial audio stream output. If zero (0) is selected, the audio stream will be output once.
- **Delete** - Allows deletion of the selected table record

[Close](#)

Audio Files Page

HELP

Audio File	Delete File
alarm.mp3	Delete
chime.mp3	Delete
patient.mp3	Delete

[Upload Audio File](#)

The Audio Files screen allows you to manage the audio files used in the [Audio Output block](#). The Audio Files screen lists all of the audio files uploaded to the SNAP system. These audio files are selectable in the [Audio Priority Management](#) table.

The Audio Files screen includes a table of uploaded audio files and the following elements:

- **Audio File** - Clicking an audio file name plays that audio file, as an audible reference
- **Delete File** - Clicking Delete for a given audio file will remove it from the SNAP system
- **Upload Audio File** - Click Upload Audio File to select an audio file for uploading to the SNAP system. Acceptable audio file formats are wav and mp3.

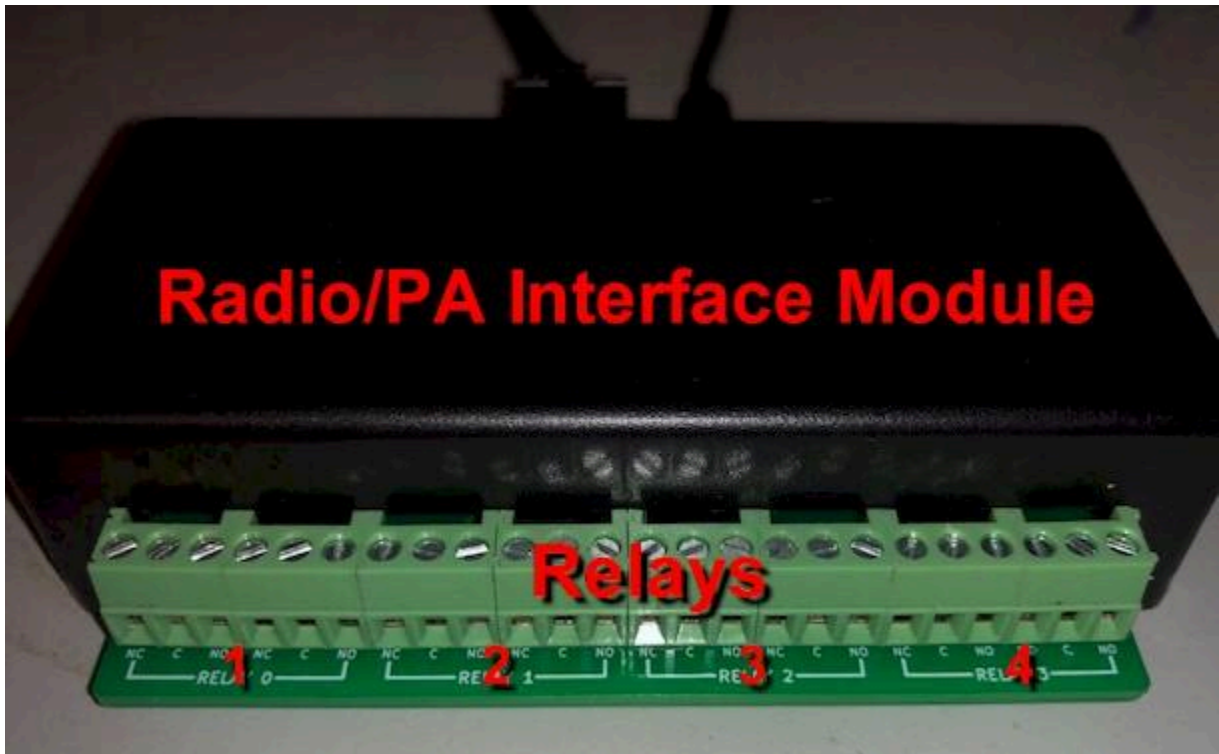
The [Audio Output block](#) uses the following operating rules:

- The output audio associated with a given alert message is formatted into one or more audio files, called an audio stream element. The audio stream elements are placed into one of four Output Queues, in a FIFO manner, based upon the Priority Level assignment rules.
- The Audio Output block will output audio stream elements from the highest priority Output Queue first, until that Output Queue is empty. Then, in sequence, lower priority Output Queues will be processed.
- When an audio stream element exists for output, and a USB Radio/PA Interface Module is attached to the SNAP system, the Audio Block will identify which PTT relay to exercise during the output, and which GPIO pin to monitor for Carrier Detect signaling, if any. If PTT relay 1 is assigned, and the associated GPIO pin is set to a value other than None, the Audio Output block will pause audio output until the Carrier Detect signal indicates the channel is available, then the Audio Output block will activate the relay and output the audio stream element the quantity of times defined by the Repeats value associated with that audio stream element, while keeping PTT relay 1 activated until the audio output stops for that audio stream element. If PTT relay 2 is assigned, as an example, and the GPIO pin value is set to None,

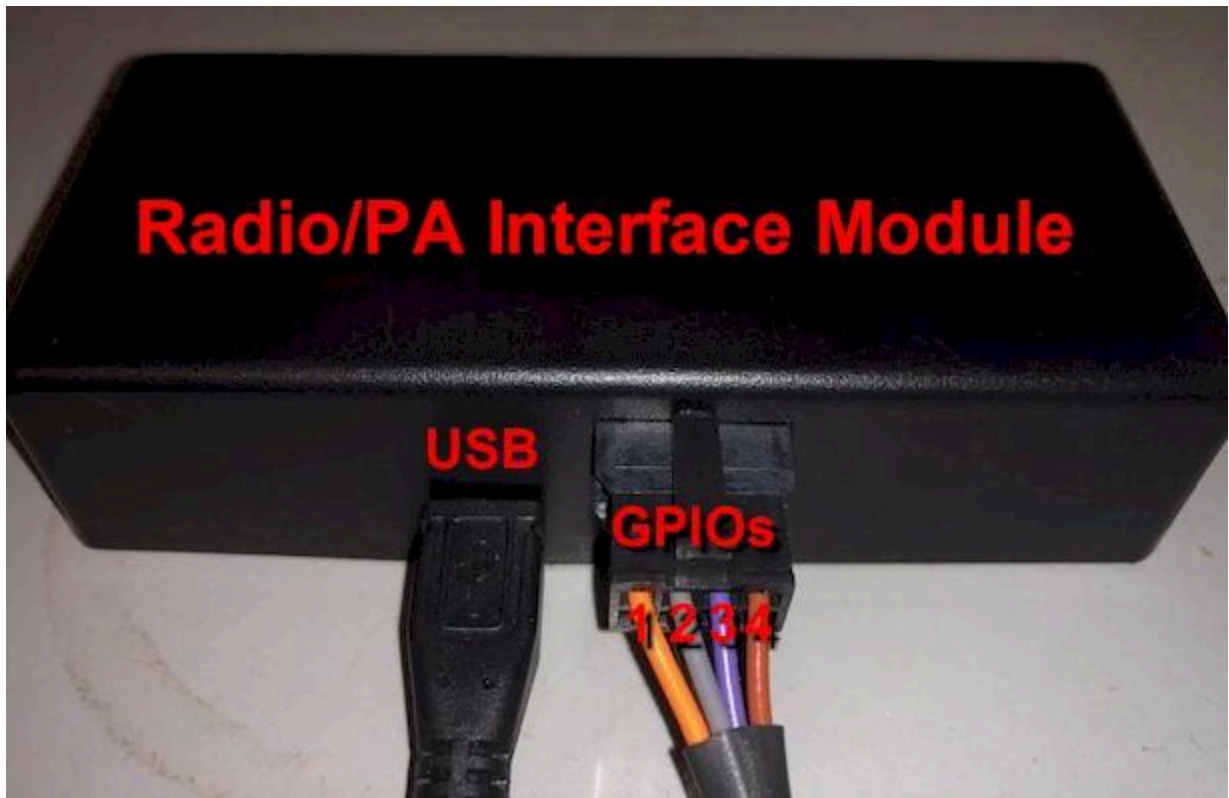
the Audio Output block will ignore the Carrier Detect signal state and will activate the relay and output the audio stream element the quantity of times defined by the Repeats value associated with that audio stream element, while keeping PTT relay 2 activated until the audio output stops for that audio stream element. If a USB Radio/PA Interface Module is not attached to the SNAP system, audio will be output without any Carrier Detect or PTT signaling.

- If a new audio stream element is added to an Output Queue, and that Output Queue is of a higher priority than the Output Queue currently being output, the audio output will be immediately halted and the higher priority Output Queue will be processed. The halted audio stream element will be output again, with the defined quantity of repeats, at the top position in its assigned Output Queue, once its assigned Output Queue starts being processed again.
- If Responder 5 Parsing is enabled, an attempt will be made to identify the status, area, room and bed parameter values in the alert message. Any status, area, room, or bed parameters identified will attempt to be associated with a matching audio file that is formatted as **area_foundarea**, where **foundarea** is the area value extracted during the parsing process. The Audio Output block is able to output either mp3 or wav file types and will attempt to locate either type when matching against the area, room and bed parameters in the Responder 5 Parsing mode. As an example, if an area value of **wing 3** is identified, an audio file named either **area_wing_3.mp3** or **area_wing_3.wav** will attempt to be located. If the associated audio file is found, it will be output.
- To use the Responder 5 Parsing feature, audio file names for the status field must be formatted with a **status_** prefix and **underscore characters instead of space characters**, e.g. for a status value of **Code Blue**, the corresponding audio file can be named **status_code_blue.wav** or **status_Code_Blue.wav** or the corresponding mp3 file type. Similar rules apply to the area, room, and bed parameters, using **area_**, **room_** and **bed_** prefixes and **underscore characters representing space characters**.
- When Responder 5 Parsing is enabled, the audio stream element will be formatted as **status+area+room+bed**, with area, room, and/or bed being optionally applied, based upon message formatting and associated audio file availability. When Responder 5 Parsing is disabled, only a single audio file will be output for the affected alert message.
- The incoming alert message value will be compared to any Key String values in the Audio Priority Management table
- If there is a key string match to the alert message value (case-insensitive), the assigned Audio File will be output in the **status** position of the output audio stream element. The audio stream element will be repeated, after the initial output, the quantity of times defined by the associated Repeats value in the Audio Priority Management table.
- If there is no key string match to the alert message value, an attempt will be made to locate an audio file named exactly as the alert message value, but using underscore characters in the audio file to represent space characters in the alert text. If an associated audio file is found, it will be output once. If an associated audio file is not found, no audio output will occur for the affected alert message.
- If there is a GPIO Pin value assigned in the Audio Priority Management table, and the assigned GPIO Pin has a pulled-to-zero contact closure event, the associated Audio File in that Audio Priority Management table record will be output, using the associated Priority Level, and repeated as defined by the Repeats value.
- The Audio Output block will queue the compiled audio stream elements into the Priority Output Queue associated with the Priority value defined in the Audio Priority Management table. If an audio stream element is created without being matched to a key string value in the Audio Priority Management table, the audio stream element will be considered the lowest priority and have a Repeats value of 0 assigned to it.

- The Audio Output block will prevent the same audio stream element to be output more than once every 30 seconds, as a means of filtering duplicate alerts arriving into the system. However, if the alert associated with the affected audio stream element becomes canceled, the duplicate output filter timer will be reset, allowing the audio stream element to be output upon detection of an associated alert with an active state.



The Radio/PA Interface Module allows the [Audio Output](#) to interface with Carrier Detect and Push-to-Talk signaling in land mobile radio and PA systems. The Radio/PA Interface Module connects to SNAP via USB port, and connects to Radio/PA systems via the GPIO interface harness and the Relays terminal block. The image above indicates relay connections, and the image below indicates GPIO and USB connections. The relays operate as NO (Normally Open), so each relay would be connected on a set of C and NO pins. The GPIO interface harness has 5 flying leads, including the 4 GPIO inputs on the top row and a black ground lead on the lower contact row. Each GPIO input is pulled to ground by shorting the input to the ground circuit of the Radio/PA Interface Module. The USB interface cable required is a USB-A male to USB-Mini-B male cable. 5VDC power is supplied to the Radio/PA Interface Module through the USB cable.



Adaptive Displays Output Protocol

```
Adaptive Displays Output [Ethernet and/or Serial]
Name: Adaptive Displays Out
Max Messages to Display: 26
Max Alert Age (sec): 300
 Timestamp Messages
Timestamp Format: AM/PM
 Hold Mode on Long Messages
 Use Display Name as Header
Header Color: White
Default Text Color: Green
IP Port: 3001
Serial Port: None
Displays Priority Mgmt Override Msgs
Modifiers:
```

The Adaptive Displays Output block encodes alert data into serial data that is delivered to compatible Adaptive (LED Display) units. The Adaptive Displays Output can control multiple display units as configured in the [Adaptive Displays table](#) and as configured in the [Adaptive Displays Priority Management table](#). Messages are truncated to

100 characters in length, as required if the alert message value is longer than 100 characters. If you need to output both serial and Ethernet data to Adaptive displays, snap a Serial Output block into the Adaptive Displays Output block.

Adaptive display models supported include:

- MNS RGB series
- 4000 RGB series
- PPD
- 215R and 215C




The Adaptive Displays Output block has the following elements:

- **Name** - Used for activity logging
- **Max Messages to Display** - The maximum quantity of active messages to be displayed on all LED displays connected to the system, ranging from 1 to 26. When the LED display message list size hits the Max Messages to Display value, new messages will replace the oldest messages in FIFO order.
- **Max Alert Age (sec)** - The quantity of seconds that a message will be allowed to be in LED display memory before it is automatically removed. Messages can also be automatically removed if the event processing system determines that an active alert event has been canceled. If the Max Alert Age value is blank or 0, the alert will be allowed to age until a cancel event is detected for that alert. The allowable values for Max Alert Age are blank and 0 through 9999.
- **Timestamp Messages checkbox** - If checked, a timestamp value will be prepended to each message, using the format defined by the Timestamp Format value
- **Timestamp Format selector** - AM/PM or 24 Hr timestamp display formats
- **Hold Mode on Long Messages checkbox** - If checked, when message lengths are longer than the visible character length of the display, the message will be presented in hold mode. Otherwise, long messages will be displayed in right to left scrolling mode. Messages shorter than the visible character length of the display always display in hold mode, regardless of this setting.
- **Use Display Name as Header checkbox** - If checked, a header value will be displayed with the message, with the header value being the Name of the associated display in the [Adaptive Displays table](#). The header will be displayed in the top line of the display.
- **Header Color selector** - The color that will be applied to the message header
- **Default Text Color selector** - Controls the text display color for normal non-priority messages
- **IP Port** - The IP Port value used for connecting to the Ethernet capable Adaptive displays in the system. IP addresses are configured in the [Adaptive Displays table](#)
- **Serial Port** - The Serial Port value used for connecting to the Adaptive displays that require serial port control. Default value is None. All serial data output for the Adaptive Displays output is hard coded to operate at 9600 baud, no parity, 8 data bits and 1 stop bit.
- **Displays button** - Clicking this button opens the [Adaptive Displays table](#)
- **Priority Mgmt button** - Clicking this button opens the [Adaptive Displays Priority Management table](#)
- **Override Msgs button** - Clicking this button opens the [Adaptive Displays Priority Override Messages table](#)
- **Modifiers** - Any Modifier blocks inserted in the Modifiers section can affect alert objects prior to their conversion to alert data for output from the Protocol block

Close

Adaptive Displays

Filter :type any text here + Add New Display [HELP](#)

Name	Destination	IP Address	Serial Address	Model	Default Message	Schedule	Delete
Call_ID_1	101		01	PPD		None	
Call_ID_2	102		02	PPD		None	
1_NE_MNS	201	192.168.10.112	00	4080 RGB or MNS	No Alerts	None	
Gift Shop	300	192.168.10.152	00	4080 RGB or MNS	No Alerts	None	

The Adaptive Displays table has the following elements:



- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Display button** - Clicking this button prompts for a Name value and creates a new table row
- **Name** - The reference Name that the Display can display at the top of the Display if the Use Display Name as Header checkbox is checked in the Adaptive Displays Output block. Also used for activity logging purposes. Note that Name values should not be longer than 11 characters for a 4080 model, 16 characters for a 4120 model, or 22 characters for a 4160 model, as the name value is displayed as boldfaced header characters on the display.
- **Destination** - The value that alert destination values must have in order to be displayed on the selected Display. If you leave this field blank, all alerts will be displayed on the selected Display.
- **IP Address** - The IP Address of the LED display, which should match the IP address programmed into the associated LED display
- **Serial Address** - The Serial Address of the LED display, which should match the Serial address programmed into the associated LED display. Serial address values must be 2 ASCII Hexadecimal digits, ranging from 00 to FF or ff. Default value is 00. If an Adaptive display is programmed with a Serial Address value other than 00, it will not display any messages other than those that include a Serial Address parameter value that exactly matches the display programming. Note that when a display uses Ethernet for control, the IP Address provides sufficient uniqueness such that you can allow all of the Ethernet controlled displays to use the default Serial Address value of 00.
- **Model selector** - The model value that will control how display header values and alert message values are displayed. The model values include:
 - 4080 RGB or MNS (13 characters wide)
 - 4120 RGB or MNS (19 characters wide)
 - 4160 RGB or MNS (26 characters wide)
 - 4200 RGB or MNS (32 characters wide)
 - 4240 RGB or MNS (39 characters wide)
 - PPD (20 characters wide)
 - 215R or 215C (15 characters wide)
- **Default Message** - The message value that will display when there are no active events or active priority messages to be displayed
- **Schedule** - Optional schedule assignment to allow processing alerts for the selected Display only when the assigned [Schedule](#) is active

- **Delete** - Allows deletion of the selected table record

Close

Adaptive Displays Priority Management

Filter :type any text here + Add New Display Priority Rule HELP

Key String	Color	Override Message	Delete
evacuate	Default	Evacuate	
clear	Green	All Clear	

The Adaptive Displays Priority Management table causes the [Adaptive Displays](#) Output block to apply priority, color or override message enhancements to displayed alerts, if Key String and/or Destination matches occur for a particular alert. If Key String and Destination fields are both blank, that table row will not be processed. If both Key String and Destination fields are non-blank, an alert must match both Key String and Destination in order for the priority, color, or override message enhancements to be applied. If an alert matches a particular table row, no more table rows will be processed for the affected alert. When the Override Message field is non-blank, when a match occurs, the Override Message will take priority over the Color assignment.







The Adaptive Displays Priority Management table has the following elements:

- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Display Priority Rule button** - Clicking this button prompts for a Key String value and creates a new table row
- **Key String** - String value that is compared to alert message values in alerts processed by the Adaptive Displays Output block. When an alert message value contains a Key String value, the associated Color or Override Message will be applied to the affected alert as it is displayed on all Displays associated with the alert that matched.
- **Color** - The Color that should be applied to the displayed text. This is applied only if the associated Override Message field value is blank.
- **Override Message selector** - If not set to None, the selected Override Message will be sent to the affected Displays associated with the alert that matched.
- **Delete** - Allows deletion of the selected table record

Close

Adaptive Displays Priority Override Messages

Filter :type any text here + Add New Priority Override Message HELP

Name	Override Message	Delete
Bomb Threat	{22}b{Blue}{12}{09}BOMB THREAT{0D}{1B}&a{White}EXIT IMMEDIATELY{0C}{04}	
Evacuate	{22}b{Red}{12}{09}EVACUATE{0D}{1B}&a{White}PROCEED TO NEAREST EXIT{0C}{04}	
EXIT	{20}b{Red}{1A}9{12}<EXIT>{0D}{1B}{22}b{Red}{12}{1A}3{09}<EXIT>{1B}&a{White}USE STAIRWELL{0D}DO NOT USE ELEVATOR{04}	
Fire Alarm	{22}b{Red}{1A}3{07}1{12}{09}FIRE ALARM{0D}{07}0{1B}{22}b{Red}{12}FIRE ALARM{1A}3{1B}&a{White}PROCEED TO NEAREST EXIT{04}	
Severe Weather	{22}b{Red}{09}SEVERE WEATHER{1A}3{1B}&a{White}TAKE SHELTER IMMEDIATELY{0C}{04}	
All Clear	{22}b{Green}{12}{09}ALL CLEAR{0D}{1B}&a{White}RETURN TO NORMAL ACTIVITY{0C}{04}	

The Adaptive Displays Priority Override Messages table allows configuration of the Override Messages used in the system. Multiple override messages can be active simultaneously.

The Adaptive Displays Priority Override Messages table has the following elements:

- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Priority Override Message button** - Clicking this button prompts for a Name value and creates a new table row
- **Name** - String value that represents the name of the Override message
- **Override Message** - The specially formatted string value that defines a specific priority override message. When the display receives the override message, that is the only message that will be displayed, until the priority override alert is canceled.
- **Delete** - Allows deletion of the selected table record

The Adaptive Displays Output uses the following operating rules:

- The primary routing/filtering method is the assignment of Destination values to displays in the Adaptive Displays table. When the Destination value of an alert matches the Destination value of one or more displays, attempts will be made to output data to Ethernet and/or Serial ports
- You can output serial data and/or Ethernet data, depending on how the block and associated tables are configured, and the requirements of the particular display. If you have a Serial Port assigned in the block, an attempt will be made to output data on the assigned serial port for any displays with matching Destination values if that record has a blank IP Address value. If the matching display has an IP Address value assigned to it, an attempt will be made to output data to the assigned IP address via Ethernet.
- To be able to have one alert be delivered to multiple displays, you would need to create multiple Adaptive Displays table records, each with the same destination value
- The Serial Address value defined in the Adaptive Displays table will be included in the formatted output message for both Ethernet and Serial port types of data output. The primary purpose of using Serial Address is to allow selective display of messages when multiple Adaptive displays are connected in a daisy-chain on the same serial data connection.
- Using the default Serial Address value of 00 causes all displays that receive that message to display that message, regardless of their Serial Address setting. Using any other Serial Address value runs the risk of the display filtering out that message upon receipt if that message does not contain a Serial Address value that matches that programmed into the display. Adaptive displays indicate their assigned Serial Address value upon power-up.

Rauland RSI Output Protocol



The image shows a configuration window titled "Rauland RSI Output [Ethernet Only]". It contains several input fields and checkboxes. The "Name" field is set to "SNAP Rauland RSI", "Vendor Name" is "SNAP", and "Max Mom Alert Age (sec)" is "999". There are three unchecked checkboxes: "Extract First/Last Names", "Last Name is First", and "Include Remainder in Last Name". Below these are five buttons: "Locations", "Action IDs", "Keystings", "Upload Config", and "Modifiers".

```
Rauland RSI Output [Ethernet Only]
Name: SNAP Rauland RSI
Vendor Name: SNAP
Vendor ID:
Consumer Key:
Consumer Secret:
RSI Server URL:
Max Mom Alert Age (sec): 999
[ ] Extract First/Last Names
[ ] Last Name is First
[ ] Include Remainder in Last Name
Locations
Action IDs
Keystings
Upload Config
Modifiers:
```

The Rauland RSI Output block encodes alert data into the Rauland RSI protocol, required to notify Rauland Responder 5000(tm) systems with externally generated alert event information.

The Rauland RSI interface requires configuration of a number of elements in order to successfully interface to the Rauland RSI server. A Rauland R5K Registration app must be used to generate some of the configuration values. The Rauland R5K Registration app is provided by Ametek.

The Rauland RSI Output block provides configuration of the Rauland RSI output and includes the following elements:

- **Name** - Used for activity logging
- **Vendor Name** - Required for successful connectivity to the Rauland RSI server. Generated using the Rauland R5K Registration app.
- **Vendor ID** - Required for successful connectivity to the Rauland RSI server. Generated using the Rauland R5K Registration app.
- **Consumer Key** - Required for successful connectivity to the Rauland RSI server. Generated using the Rauland R5K Registration app.
- **Consumer Secret** - Required for successful connectivity to the Rauland RSI server. Generated using the Rauland R5K Registration app.
- **RSI Server URL** - The URL pointing to the Rauland RSI server. The URL can optionally include an HTTP:// or HTTPS:// prefix.

- **Max Mom Alert Age (sec)** - Defines the maximum alert aging allowed for alerts generated that have a momentary state. Rauland RSI requires alerts to typically be notified of both start and stop events. The Max Mom Alert Age value allows the SNAP system to auto-generate an alert stop event when the age of a momentary alert reaches the Max Mom Alert Age value, in seconds. The value can range from 0 to 999. A value of zero (0) causes the system to not auto-generate an alert stop event on a timer. The default value is 999.
- **Extract First/Last Names checkbox** - If checked, after Location and Action ID are extracted from an alert message value, the remainder of the message can be output as metadata, using both the First Name and Last Name fields, to the Rauland RSI API (see [Rauland RSI Output operating rules](#)).
- **Last Name is First checkbox** - If checked, and the remainder of the message can be output as metadata, the First and Last Name extraction method will assume that the first word of the remainder is the Last Name instead of the First Name.
- **Include Remainder in Last Name checkbox** - If checked, and the remainder of the message can be output as metadata, if multiple words are available for output in the Last Name field, all of the words will be used in the Last Name field instead of just the first word. If the Extract First/Last Names checkbox is not checked, this checkbox has no effect on system operation.
- **Locations button** - Clicking this button opens the [Rauland RSI Locations table](#)
- **Action IDs button** - Clicking this button opens the [Rauland RSI Action IDs table](#)
- **Keystings button** - Clicking this button opens the [Rauland RSI Keystings table](#)
- **Upload Config button** - Clicking this button opens the [Rauland RSI Upload Configuration page](#)
- **Modifiers** - All Modifier blocks inserted in the Modifiers section can affect or examine alert objects prior to their output from the Protocol block

Rauland RSI Output [Ethernet Only]

Name: SNAP Rauland RSI

Vendor Name: ASCCI

Vendor ID: 823

Consumer Key: A773s22JY17rtu66n9k1jaskkkd89i883kdjkikkwiaiskas...

Consumer Secret: 883kdiwksiiake9diksiksk1idksoosd9a

RSI Server URL: 192.168.10.241

Max Mom Alert Age (sec): 999

Extract First/Last Names

Last Name is First

Include Remainder in Last Name

Locations

Action IDs

Keystings




Upload Config

Modifiers:

Close

Rauland RSI Locations

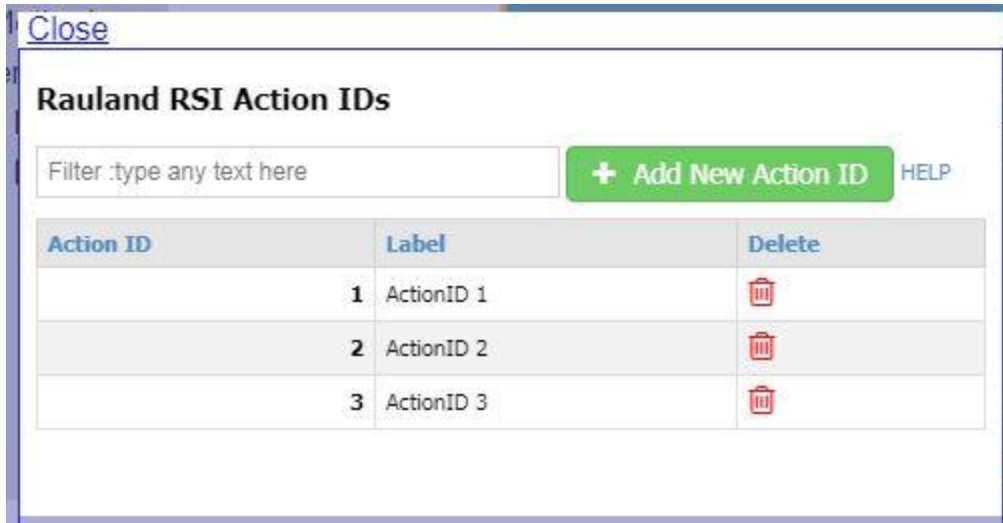
Filter :type any text here + Add New Location HELP

Location	Delete
101:101:101	
101:101:102	
101:101:103	

The Rauland RSI Locations table is used for creating and maintaining a list of Location values to be passed with alert registration events to the Rauland system. The Location values are assigned using a dropdown selector in the Rauland RSI Keystings table.

The Rauland RSI Locations table has the following elements:

- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Location button** - Clicking this button creates a new table row
- **Location** - A string value that is usually created using an Area:Room:Bed display format
- **Delete** - Allows deletion of the selected table record



The Rauland RSI Action IDs table is used for creating and maintaining a list of Action ID values to be passed with alert registration events to the Rauland system. The Action ID values are assigned using a dropdown selector in the Rauland RSI Keystrings table.

The Rauland RSI Action IDs table has the following elements:

- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Action ID button** - Clicking this button creates a new table row
- **Action ID** - An integer value that causes the Rauland RSI Output block to assign that Action ID value to any alert whose message value includes the Key String value associated with the specific Action ID.
- **Label** - A text value that will be used as a reference in the dropdown selector in the Rauland RSI Keystrings table.
- **Delete** - Allows deletion of the selected table record

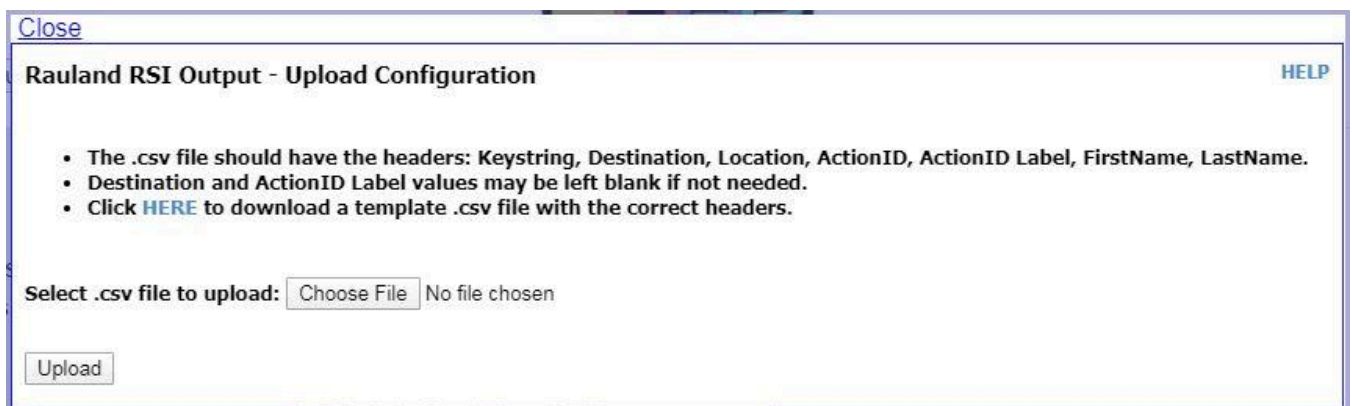
Rauland RSI Keystrings

Keystring	Destination	Location	Action ID	First Name	Last Name	Delete
A101		A101	None			
A102		A102	None			
A103		A103	None			
IT ROOM		None	IT ROOM			
KITCHEN		None	KITCHEN			
RESTROOM		None	RESTROOM			
John Doe		None	None	John	Doe	
Freddy		None	None	Freddy		
Krueger		None	None		Krueger	

The Rauland RSI Keystings table causes the [Rauland RSI](#) Output block to be able to use key string and destination value detection as a method of deciding which alerts to output and which Action ID and Location values to assign to outgoing alerts.

The Rauland RSI Keystings table has the following elements:

- **Filter** - Entering a Filter value reduces the visible table rows to those that match the filter
- **Add New Keysting button** - Clicking this button creates a new table row
- **Keystring** - A string value that is compared to alert message values in alerts processed by the Rauland RSI Output block, and is case-insensitive. When an alert message value contains a Keystring value, the associated Action ID value will be assigned to the associated alert. The alert will be output to Rauland if it also meets the requirements of the Destination field below. If the Keystring field and the Destination field are both blank, no alerts will be output for that Keystings table record.
- **Destination** - A string value that, if non-blank, filters out alerts unless the Destination value of a given alert matches the Destination value of the Keystings table record. If the Destination field is blank, only the Keystring field will be used to compare to an alert for matching.
- **Location selector** - A selector that allows assignment of a Location value to any alerts that match the Keystings record. If the Location value is set to None, no alerts will be output for that record.
- **Action ID selector** - A selector that allows assignment of an Action ID value to any alerts that match the Keystings record. If the Action ID value is set to None, no alerts will be output for that record.
- **First Name** - A string value that, if non-blank, and if the Keystring value matches an alert message, and if the message remainder after extraction of Location and Action ID values is non-blank, will be delivered via the Rauland RSI API as the First Name field.
- **Last Name** - A string value that, if non-blank, and if the Keystring value matches an alert message, and if the message remainder after extraction of Location and Action ID values is non-blank, will be delivered via the Rauland RSI API as the Last Name field.
- **Delete** - Allows deletion of the selected table record



Close

Rauland RSI Output - Upload Configuration HELP

- The .csv file should have the headers: Keystring, Destination, Location, ActionID, ActionID Label, FirstName, LastName.
- Destination and ActionID Label values may be left blank if not needed.
- Click [HERE](#) to download a template .csv file with the correct headers.

Select .csv file to upload: No file chosen

The Rauland RSI Upload Configuration page allows the optional uploading of a specially formatted CSV file, to allow immediate population of the Locations, Action IDs and Keystings tables. Uploading a CSV file will overwrite all values that were previously in those 3 tables. Note that you can choose to manually edit all 3 tables from within each table, but any edits will be replaced by any CSV file that you successfully upload.

The Rauland RSI Upload Configuration page has the following elements:

- **Download template CSV file link** - Clicking this link allows you to download/open a specially formatted CSV file that allows you to build a list of all of the records that would go into the Locations, Action IDs and Keystings tables. Please note that when you format **Location** values using numbers and colon (:) delimiters, the Excel spreadsheet will auto-format those values into decimal numbers unless you change the Cell Formatting for the Location column to **Text**. **ActionID Label** values are optional, and are used to populate the Action IDs table. If you choose to include ActionID Label values in your CSV file, note that only the first occurrence of a non-blank ActionID Label for a given Action ID value will be recorded in the Action IDs table. This means that you do not need to duplicate ActionID Label values throughout the CSV file. The CSV file template appears as follows when opened in a spreadsheet:

	A	B	C	D	E	F	G
1	Keystring	Destination	Location	ActionID	ActionID Label	FirstName	LastName
2	Test Alert		2:44:03	1	Nurse Call		
3							
4							

- **Choose File button** - Clicking the Choose File button allows you to select your filled out CSV file for uploading.
- **Upload button** - Clicking the Upload button uploads the selected CSV file and tries to immediately populate the Locations, Action IDs and Keystings tables with data. If the upload fails, you would most likely need to edit the CSV file headers per the notes on the Upload Configuration page.

You can test your Rauland RSI Output configuration against an RSI Server Simulator built into the SNAP system. To connect to the RSI Server Simulator, configure fields in the Rauland RSI Output block as shown below:

- **Vendor ID** - 1
- **Consumer Key** - CONSUMERKEY
- **Consumer Secret** - CONSUMERSECRET
- **RSI Server URL** - The root URL of the SNAP system, e.g. 192.168.10.247

Rauland RSI Output [Ethernet Only]

Name: SNAP Rauland RSI

Vendor Name: SNAP

Vendor ID: 1

Consumer Key: CONSUMERKEY

Consumer Secret: CONSUMERSECRET

RSI Server URL: 127.0.0.1

Max Mom Alert Age (sec): 999

Extract First/Last Names

Last Name is First

Include Remainder in Last Name

Locations

Action IDs

Keystings

Upload Config

Modifiers:

You can observe the SNAP system log file to see if registration to the server was successful, as well as observation if any alerts were output by the Rauland RSI Output block.

The [Rauland RSI Output block](#) uses the following operating rules:

- A matching criteria is used to determine which alerts to send to the Rauland system, using the following rules.
 - Any active alert is compared to both the Keystring and/or Destination field values in each record of the Keystings table. The Message field value of the alert is compared to the Keystring value in the record and the Destination field value of the alert is compared to the Destination value in the record.
 - If the Keystring value of a particular record is contained in the Message value of an alert as a substring, case-insensitive, the alert is a notification candidate for that record
 - If the Destination value of a particular record includes a value that matches the Destination value of an alert, the alert is a notification candidate for that record
 - If the Keystring value is non-blank and the Destination value is blank, for a table record, only the Keystring matching rule will be used to determine whether an alert should be a candidate for delivery to the Rauland system
 - If the Keystring value is blank and the Destination value is non-blank, only the Destination matching rule will be used to determine whether an alert should be a candidate for delivery to the Rauland system
 - The definition of a full matching alert is when both a Location and Action ID value are defined for a given alert. A full matching alert can also include First Name and/or Last Name values if

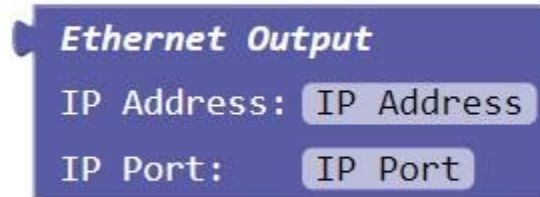
keystring matches occur. For any given alert, any record in the Keystrings table that identifies either Location or Action ID will be compared to identify the associated value used to format the alert notification.

If multiple records in the Keystrings table can be used to define any of the parameter values used to define an alert for output, the following prioritization rules will apply:

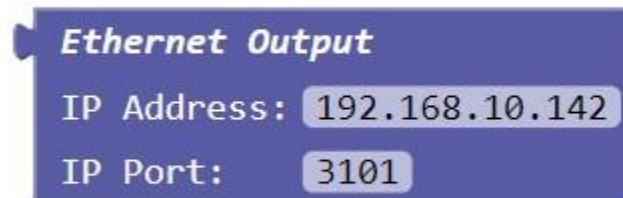
- Keystring match closest to beginning of alert message
- Destination value match
- Both Location and Action ID may be defined in a single Keystrings table record, but may also be defined in two different records. In addition, the First Name and Last Name field values can be defined in the same record, combined with a Keystrings record that contains a Location and/or Action ID value, or on a separate Keystrings record. As matching occurs, the first instance of a non-blank First Name field value and the first instance of a non-blank Last Name field value will be used to format the outgoing alert. Once a full match is made, no other records will be examined for that alert.
- When an alert event meets the matching criteria, if the alert state is active, the Rauland RSI Output block will notify the Rauland RSI server of the alert activation, using the associated Action ID and Location values defined in the Rauland RSI Keystrings table for the matching record.
- When an alert event meets the matching criteria, if the alert state is momentary, the Rauland RSI Output block will notify the Rauland RSI server of the alert activation, using the matching Action ID and Location values defined in the Rauland RSI Keystrings table. Any momentary alert events can also be age monitored (using the **Max Mom Alert Age** value) so that an automatic alert cancellation event can be generated and sent to the Rauland RSI server when the momentary alert age exceeds the Max Mom Alert Age value defined in the Rauland RSI Output block.
- When an alert event is detected, if the alert state is canceled, the Rauland RSI Output block will notify the Rauland RSI server of the alert cancellation event
- If the **Extract First/Last Names** checkbox is unchecked, First Name and/or Last Name values will only be output for a given alert if they are identified while processing the alert through the Keystrings table.
- If the **Extract First/Last Names** checkbox is checked and there is a remainder value from the alert message after Action ID and Location values have been extracted, and neither a First Name nor a Last Name value were identified while processing the alert through the Keystrings table, the alert message remainder value will attempt to be split to pass values to both the First Name and Last Name parameters. The default split method is that the first word will be used for First Name value and either the balance of the alert message remainder will be used for the Last Name portion or just the first word of the balance of the alert message remainder will be used, depending on the state of the **Include Remainder in Last Name** checkbox. For example, if the original alert message value was **C301 John Doe PENDANT EXTRA DATA** and the message remainder was **John Doe EXTRA DATA**, the First Name field would display as **John** and the Last Name field would display as **Doe EXTRA DATA** if the **Include Remainder in Last Name** checkbox was checked. Otherwise, the Last Name field value would display as **Doe**. The First Name field value would always be output as a single word, with the Last Name field containing one or more words of the balance.
- If the **Last Name is First** checkbox is checked the Extract First/Last Names method defined above would use the first word in the remainder as the Last Name value. For example, if the original alert message value was **C301 Doe John PENDANT EXTRA DATA** and the message remainder was **Doe John EXTRA DATA**, the First Name field would display as **John** and the Last Name field would display as **Doe EXTRA DATA** if the **Include Remainder in Last Name** checkbox was checked. Otherwise, the Last Name field value would display as **Doe**.

Output Methods

Ethernet Output Method



The Ethernet Output Method block is snapped into appropriate Output Protocol blocks, to define an Ethernet method of data collection. Edit the IP Address and IP Port values appropriate to the associated Output Protocol block, and the SNAP server will attempt to make the appropriate Ethernet connections to the assigned IP Address and IP port.



Serial Output Method



The Serial Output Method block is snapped into appropriate Output Protocol blocks, to define a serial port method of data output. Edit the Port, Settings and Flow Control values appropriate to the associated Output Protocol block, and the SNAP server will output serial data at the assigned settings.

Capture Output Method



The Capture Output Method block is snapped into appropriate Output Protocol blocks, and is designed for system initial configuration and troubleshooting purposes, to allow you to collect alert data output by the associated Output Protocol block into a log file viewable in a browser window.

To view the collected alert data output, click on the ellipsis button ... in the Capture Output block, which will open a web page containing data where control characters are converted to a human readable form. You can refresh the page to view accumulating data. Following is example COMP2 output data:

```
Test Output: 101<CR><LF>Test message 001<CR><LF>
```

```
Test Output: 101<CR><LF>Test message 002<CR><LF>
```

Desktop Shortcuts

User logins can be set up as desktop shortcuts, where the user double clicks to launch a connection to a specific configuration table. Following are shortcut URL examples for each of the configuration tables, using a password value of **mypw**. You would edit the server IP address and the password value.

Configuration Table	Shortcut URL Example
Accutech Door Zones	http://192.168.10.142/table?type=accutech_gateways&user=user&password=mypw
Accutech Tags	http://192.168.10.142/table?type=accutech&user=user&password=mypw
Adaptive Displays	http://192.168.10.142/table?type=mns_displays&user=user&password=mypw
Adaptive Displays Priority Management	http://192.168.10.142/table?type=mns_priority_mgmt&user=user&password=mypw

Adaptive Displays Priority Override Messages	http://192.168.10.142/table?type=mns_priority_msgs&user=user&password=mypw
Alerts List Priority Management	http://192.168.10.142/table?type=alerts_list_priority_mgmt&user=user&password=mypw
Audio Files	http://192.168.10.142/table?type=audio_files&user=user&password=mypw
Audio Output Priority Management	http://192.168.10.142/table?type=audio_priority_mgmt&user=user&password=mypw
E-mail Recipients	http://192.168.10.142/table?type=smtp_recipients&user=user&password=mypw
Filter Tables	http://192.168.10.142/table?type=filter&user=user&password=mypw
Inovonics Transmitters	http://192.168.10.142/table?type=inovonics&user=user&password=mypw
Inovonics Receivers	http://192.168.10.142/table?type=inovonics_receivers&user=user&password=mypw
Manual Cancel - Alerts List Output	http://192.168.10.142/manual_cancel_alo?user=user&password=mypw
Messaging Recipients	http://192.168.10.142/table?type=messaging_recipients&user=user&password=mypw
Monitor Tables	http://192.168.10.142/table?type=monitor&user=user&password=mypw
Pager App Priority Management	http://192.168.10.142/table?type=pager_app_priority_mgmt&user=user&password=mypw
Predefined Messages	http://192.168.10.142/table?type=predefined_msgs&user=user&password=mypw
Route Tables	http://192.168.10.142/table?type=route&user=user&password=mypw
Translation Tables	http://192.168.10.142/table?type=translate&user=user&password=mypw
Rauland RSI Locations	http://192.168.10.142/table?type=rauland_rsi_location&user=user&password=mypw
Rauland RSI Action IDs	http://192.168.10.142/table?type=rauland_rsi_actionid&user=user&password=mypw

Rauland RSI Keystings	http://192.168.10.142/table?type=rauland_rsi_keywords&user=user&password=mypw
Individual Encryption Keys	http://192.168.10.142/table?type=encryption_keys&user=user&password=mypw