# VistaLINK® PRO Auto Response



**EVERTZ MICROSYSTEMS LTD.**
5288 John Lucas Drive,
Burlington, Ontario,
Canada L7L 5Z9
Phone: 905-335-3700
Sales: sales@evertz.com Fax: 905-335-3573
Tech Support: service@evertz.com Fax: 905-335-7571
Web Page: http://www.evertz.com

Version 1.0 May 2008

*This page left intentionally blank*

# TABLE OF CONTENTS

## Figures

# REVISION HISTORY

| REVISION | DESCRIPTION | DATE |
|---|---|---|
| 1.0 | First Release | May 08 |

*This page left intentionally blank*

# 1. OVERVIEW

## 1.1. INTRODUCTION

Auto Response is an added value option to the VistaLINK® system for running automated profiles. The model number for the Auto Response option is the +SCH indication. The VistaLINK® Server can provide information on whether the option is enabled in the license by checking the virtual Auto Response system LED. The green light indicates that the option is enabled in the VistaLINK® license. See Figure 1-1.

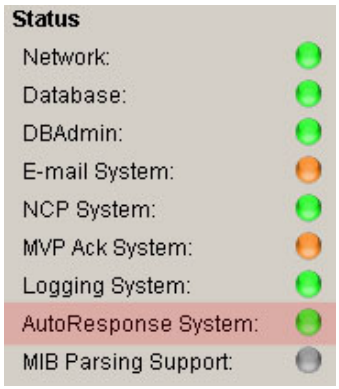



**Figure 1-1: Auto Response Enabled from the VistaLINK® License**

*Auto Response* is a system that supports many strategies and can be tailored to a variety of work environments. The purpose of *Auto Response* is to create automated systems. *Auto Response* provides all the necessary tools to create automated profiles through a GUI (Graphical User Interface). The program can manage multiple automated profiles at once. Automated profiles can be configured and maintained from any VistaLINK® Client program.

The *Auto Response* program functions similar to the VistaLINK® PRO Client. It is recommended that it be installed on a dedicated machine with redundancy features. It is important that the *Auto Response* client is operating properly so that the automated profiles can function. *Auto Response* has a built-in SNMP Agent, which is used so that remote VLPRO clients can use the *Auto Response* server for advanced alarm management.

The SNMP Agent disables Auto Response on the following types of systems:

- MVP System Manager with the SNMP option enabled
- VistaLINK® CCA Server
- Any computer systems setup with a SNMP Agent (HP Insight Agent for example)

## 1.2. CORE COMPONENTS

*Auto Response* is a system that supports many strategies. It is composed of tools that loosely tie into each other to create a custom *Auto Response* profile. The tools are broken down into the following three main categories: Triggers, Guard States, and Payloads.

### 1.2.1. Triggers

The purpose of a Trigger is to initiate the auto response profile. Triggers can be executed in the following three different ways:

- **Alarm Trigger:** Device alarm used to initiate the firing of the trigger. It is possible that multiple alarms from multiple devices can be composed into a single trigger.

- **Calendar Trigger:** A predefined scheduled time for when the trigger should be fired. Added control is given so that the fire mechanism can be re-occurring on a daily, weekly or monthly schedule. It is possible that a trigger can be configured to fire every few seconds.

- **Manual Triggers:** Triggers can be built to have no fire mechanism at all. The purpose for this strategy is to manually invoke the trigger through a Macro.

### 1.2.2. Guard States

*Guard States* are an optional component of the *Auto Response* profile. When a trigger is fired for a profile, the 2nd step to the profile execution is to evaluate the *Guard State*. Guard states provide a fork path to direct the profile execution towards one of the two directions it can take. The guard state can be evaluated to be TRUE or FALSE. The Auto Response profile can be configured to execute a configuration based on either of the evaluation states.

*Guard States* are usually made out of hardware parameters. By latching profiles onto parameters, it is possible to intelligently run a configuration based on how the card is configured or what it detects. Multiple Guard States and parameters can be composed together using logical ***AND's*** and ***OR's***. A ***NOT*** operator can be used to check the reverse status of a *Guard State*. This is commonly used to re-use existing Guard States.

Granular control is given at different parameters to match values with operators similar to:

- Equals
- Not Equals
- Greater Than
- Greater Than or Equals
- Less Than
- Less Than or Equals

A common strategy with *Guard States* is to have a TRUE path to run a specific configuration and a FALSE path to not run any configuration. The strategy behind this is to have the profile execute fully only in special cases (whether the guard state passes). It is often desired to have no *Guard State* specified in the Auto Response profile when doing simple event based triggers (converting GPI's to run custom batch configurations).

### 1.2.3. Payloads

*Payloads* are the last step to an *Auto Response* profile. Payloads are physically Macro's that run various functions. The term *Payloads* is a symbolic link to the word 'Macro'. *Payloads* can be created to make device configurations, custom TRAP alarms or to execute another profile. The success in any Auto Response profile is based on the success of the execution of the *Payload*. Typically the VistaLINK® PRO Server executes the contents of the payload when Auto Response initiates the Payload.

The three core components are grouped together to create a single Auto Response profile, however, it is possible to mix and match the elements to different profiles. For instance, a Guard State can belong to any other profile at the same time. This helps streamline the work without having to make duplicate entities.

## 1.3. AUTO RESPONSE EXAMPLES

The following flow charts represents possible Auto Response profiles. The techniques used to make them are outlined in this manual.

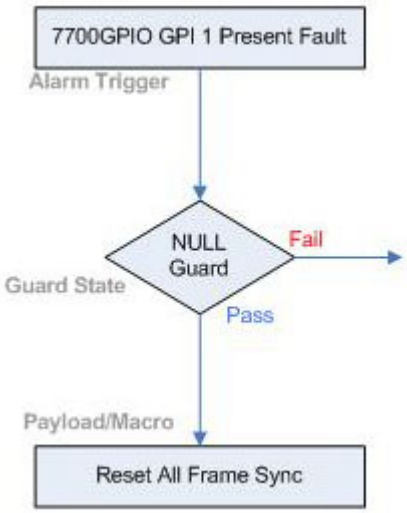



**Figure 1-2: Simple Auto Response for GPI Triggering**

The above example is a simple Auto Response profile that is composed of an alarm trigger and a macro. The trigger is fired when the system detects a GPI 1 Present fault from the 7700GPIO card. This type of response does not use a guard state so the profile always evaluates TRUE automatically. The payload configured in this profile is a macro that will reset all the Frame Syncs to factory default. The alternate payload (FALSE guard evaluation) is not configured.

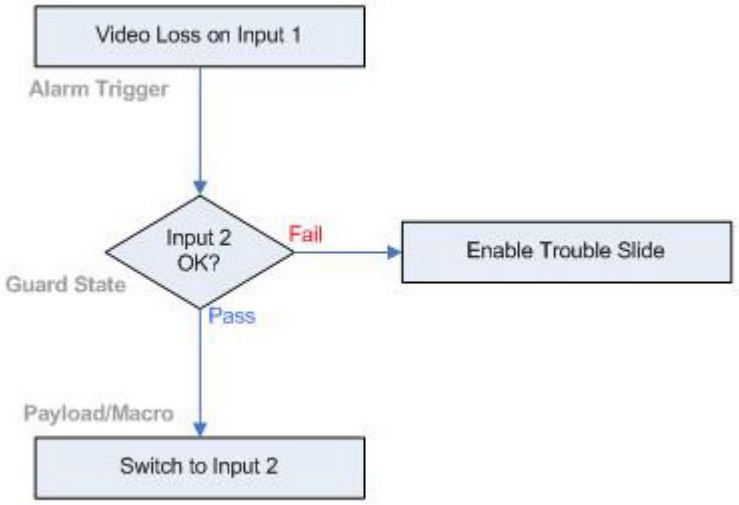



**Figure 1-3: Auto Response Profile with a Guard State**

The above Auto Response profile utilizes a guard state to intelligently determine the action. The trigger is composed of an alarm for "Loss of Input 1". When the trigger is fired it will run the Guard State configured to match a value that determines if Input 2 is present. A status indicator would be used with an EQUAL's operator. If the status indicator equals a good value, than the TRUE payload is executed. This payload would be a macro that would move the PGM path into input 2 of the switch. If the status indicator does not equal the desired value, then the guard would evaluate false. The alternate payload (FALSE evaluation) in this example would be executed. This macro would configure the PGM path to output a trouble slide.

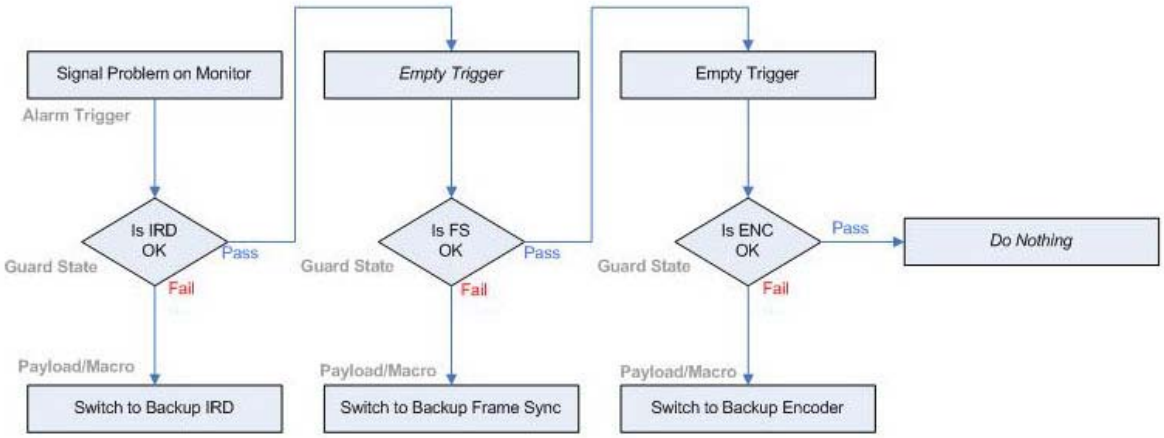



**Figure 1-4: Daisy Chained Auto Response Profiling**

The above example uses payloads to execute another Auto Response profile. The payloads are macros with a component to manually fire a trigger. The system works by detecting faults using an AVM Monitor.

If a fault alarm occurs, the trigger is fired that runs an Auto Response profile, which evaluates the first device in the signal path. If the IRD check is OK then the second Auto Response profile is fired. The purpose of this configuration is to find the faulty input and make the proper redundant change over for the point that has the problem.

Faults can occur but do not necessarily indicate that a device has a faulty input. When the response profiles execute, it is expected that it would finish at the last response, which would do nothing. The key to using Guard States in most Auto Response profiles is to prevent the payload from executing all the time.

# 2. INSTALLATION

## 2.1. INSTALLATION OF THE VISTALINK® AUTO RESPONSE

### 2.1.1. Component Check-List

Before operating the VistaLINK® monitoring suite, your machine must meet the following requirements:

- Pentium 4 1.0 GHz (or higher)
- 512MB RAM
- 100Mb Ethernet Card, TCP/IP configured
- 8MB Video Card
- 1024X768 screen resolution
- Windows NT4, 2000,XP, Server 2003 operating system
- CD-ROM
- Evertz Monitoring Toolkit CD

### 2.1.2. Initiating the Software Installation for the Auto Response Client

When installing your first client, it is recommended that it be installed on the same machine as the VistaLINK® PRO Server.

To start the client installer, navigate to the Monitoring Toolkit CD splash screen and click the 'Install VistaLINK® Client (30 Day Trial / Full Version)' link.

Use the Monitoring Toolkit CD to install clients on remote computers.


**Figure 2-1: VistaLINK® Monitoring Toolkit Screen**

### 2.1.3. VistaLINK® PRO Auto Response Installer Welcome Screen

After the installer finishes extracting its contents, the install script starts with a note that informs the user to shut down any programs that may be running. Select the 'Next' button when ready.

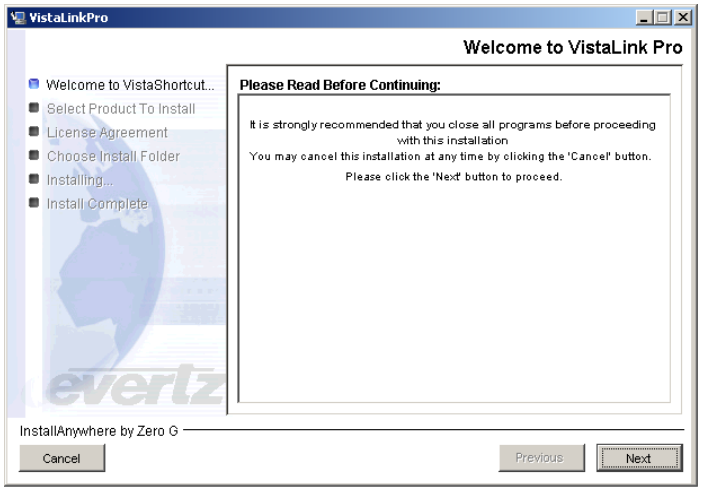



**Figure 2-2: Installer Welcome Screen**

### 2.1.4. Client Selection Menu

This menu allows the user to select a particular VistaLINK® PRO client to install. The remainder of this manual will refer to "VistaLinkPro AutoResponse". Select the "VistaLinkPro AutoResponse" graphics button and click the 'Next' Button to advance to the next menu.

***Note: The other client types have the same installation and setup method. Please refer to your VistaLINK® PRO Server license information.

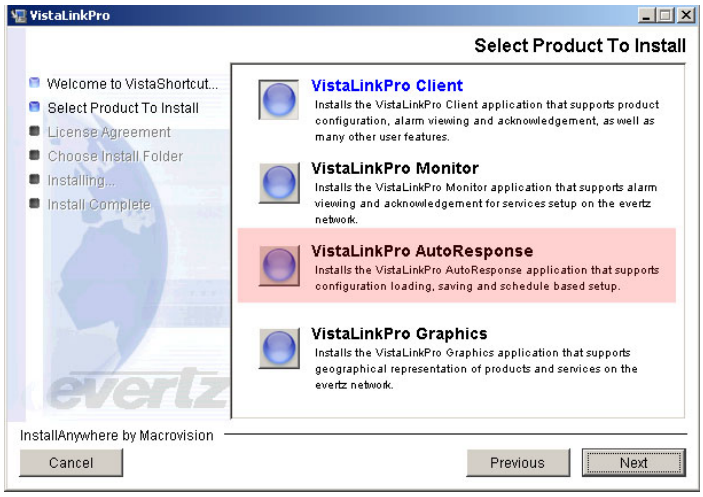



**Figure 2-3: Client Selection**

### 2.1.5. Select the Installation Folder

From the installation folder menu, a custom installation path can be made with either the text field or the 'Choose…' button. A note is provided about existing VistaLINK® PRO Auto Response installations. If the 'Installation Folder' path contains an existing VistaLINK® PRO AutoResponse installation, the install script will write over top of it. Select the 'Install' button when you are ready to start installing the application.

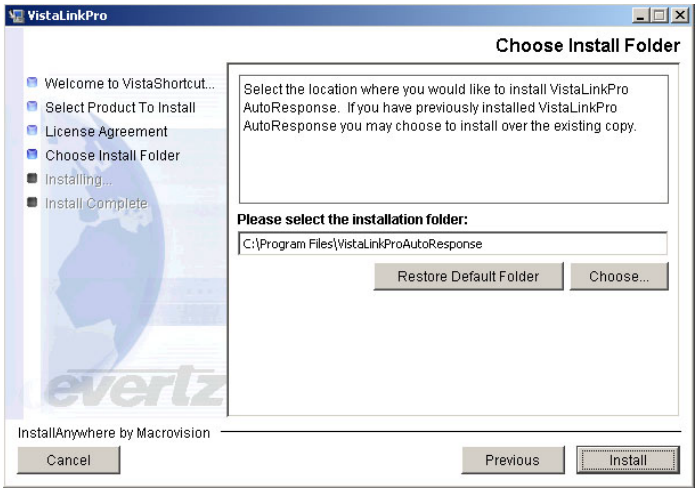



**Figure 2-4: Install Folder**

### 2.1.6. Installation Complete

Once the VistaLINK® PRO client has been successfully installed, select the 'Done' button to quit the installer.

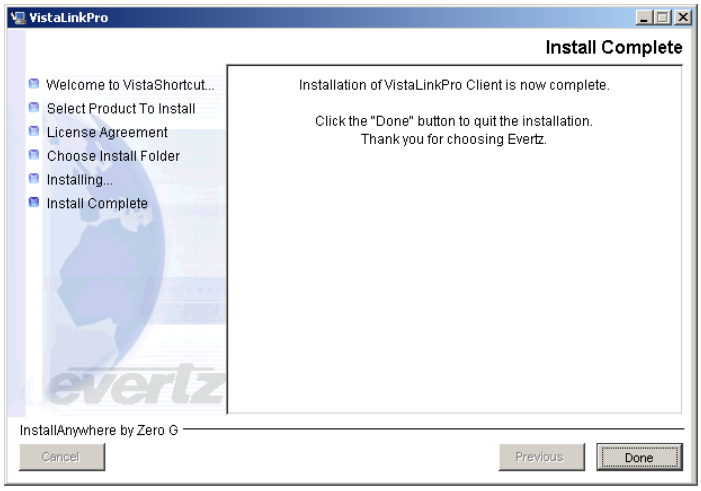



**Figure 2-5: Install Complete Screen**

## 2.2. ENABLING AUTO RESPONSE TREE NODES

By default, the *Auto Response* tree nodes are not enabled in the Navigation Tree. To enable the *Auto Response* nodes click 'Tree' and go to the 'Properties' option from the top drop down menu of the application.

*Figure 2-6* highlights which nodes are needed to build *Auto Response* profiles.

The four tree nodes needed for Auto Response are:

1. Guard States
2. Macros
3. Responses
4. Triggers

It is possible to manage Auto Response from any other type of VistaLINK® PRO Client. Ensure that the other Clients have the Auto Response tree nodes enabled from their **Tree Properties**.

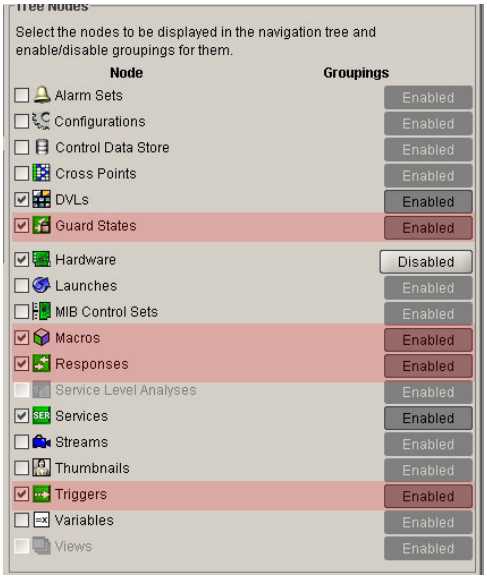



**Figure 2-6: Tree Properties**

**These settings will be remembered when the Client is restarted, however, the settings are not global to all Clients. It is a local configuration setting only.**

The *Auto Response* tree nodes can have two types of states. These states represent whether the *Auto Response* client is running or not. Figure 2-7 and Figure 3-2 show that the *Auto Response* tree nodes are unverified (gray) when the client is not running.


**Figure 2-7: Unverified Auto Response**


**Figure 2-8:Verified Auto Response**

*Auto Response* runs a built in SNMP agent that is discoverable from any VistaLINK® Client. A good practice is to verify that the *Auto Response* server is discoverable in the Hardware Tree from a client. The *Auto Response* hardware icon is represented through this standard icon .

**Note: If more than one *Auto Response* Client is running on the network for other VistaLINK® Systems, they will also be shown in the Hardware Tree. It is not possible to determine which *Auto Response* Client is connected to which VistaLINK® Server from the Hardware Tree.**

## 3. TRIGGERS

Building any type of *Auto Response* profile starts with deciding on what the fire mechanism for the profile should be. Triggers provide the tools to create this fire mechanism. Triggers are built and maintained using the Navigation Tree from either the Auto Response client or any other VistaLINK® Client. The following icon represents the Triggers node .

To create a new Trigger, right click the *Triggers* option from the Navigation Tree, then select 'New' followed by the 'Trigger' option.

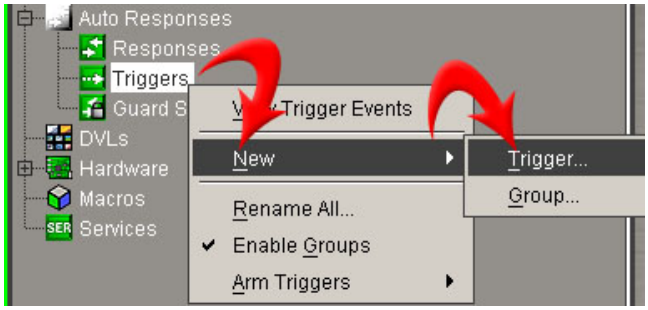



**Figure 3-1: Using the Navigation Tree to Create a Trigger**

Once selected, the *Trigger Editor* will open, as shown in Figure 3-2.

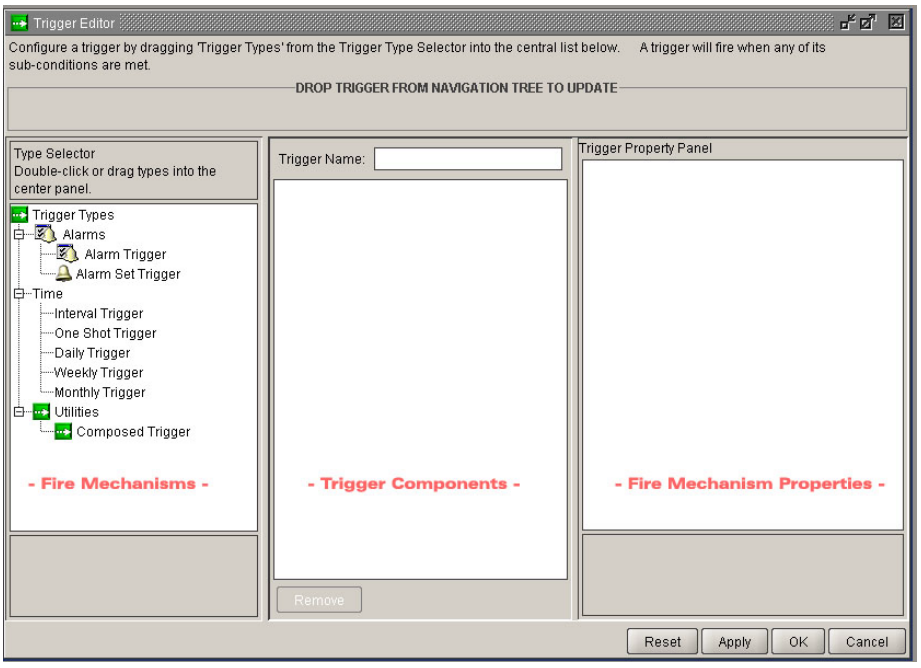



**Figure 3-2: Trigger Editor**

The Trigger Editor is composed of three columns that provide a work flow from left to right. The left most column is used to select the trigger's fire mechanism. The desired trigger components used would be placed into the middle column. Since a Trigger can have multiple types of fire mechanisms, multiple entries can exist in the middle column. The right side column is the properties column for the fire mechanism that is selected in the middle column. The properties column updates automatically to show the correct properties for whichever fire mechanism is selected from the middle column.

## 3.1. ALARM TRIGGERS

Alarm Triggers is a mechanism to fire the trigger based on a device alarm. To use an Alarm Trigger, drag and drop the  Alarm Trigger component from the left column to the middle column of the editor. The editor is then updated to have an item in the middle column and the properties shown in the right most columns.


**Figure 3-3: Alarm Based Trigger Properties**

Alarm Triggers have four properties for configuration. The following is a list of the properties:

- **Hardware:** Used to configure which device from the Navigation Tree should contribute to firing the Trigger. Click the 'Click to Edit' value and a "**…**" button will appear. Select the "**…**" button to open up the Hardware Selector dialog. Select a device from the hardware tree and click 'OK'.

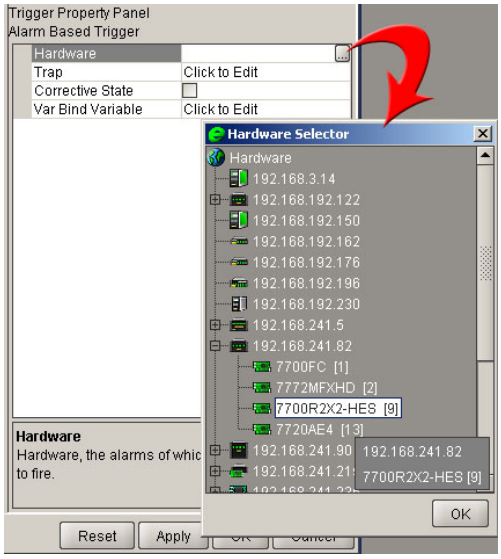



**Figure 3-4: Hardware Selector Dialog**

- **Trap:** Used to configure which trap would contribute to firing the trigger for the hardware device selected from above. Click the 'Click to Edit' value and a "**…**" button will appear. Select the "**…**" button and the Alarm Selector dialog will appear. The Alarm Selector dialog will automatically show all the alarms for the hardware device selected in the 'Hardware' parameter. Select an alarm from the list and click 'OK'.

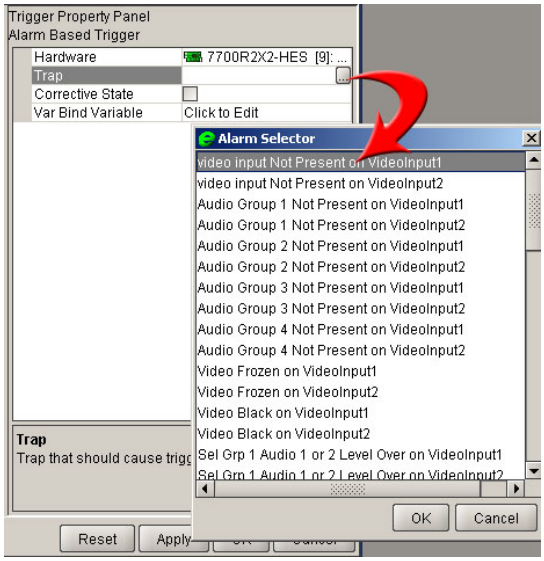



**Figure 3-5: Alarm Selector Dialog**

- **Corrective State:** The Corrective State parameter is used to enable the correction alarms for the alarm selected in the trap. For example, in Loss of Video, selecting the corrective state for that alarm would indicate that the alarm is sent when the video has regained connection to the device. This parameter is not mandatory to configure.

- **Var Bind Variable:** This parameter specifies the index variable used to store the alarms variable binding data for later processing and analysis. This parameter is not mandatory to configure.

Once the Trigger has been fully configured, ensure that the Trigger has a descriptive label entered in the 'Trigger Name' field. Click 'Apply' to save the newly created trigger. The trigger would then appear with a descriptive label in the Triggers node of the Navigation Tree.


**Trigger Name Text Field**

**It is possible to have a trigger built up with multiple Alarm fires employing a variety of devices. The middle column can support an unlimited amount of items.**

**For more efficient use, drag a device from the Hardware tree into the Trigger editor. The Alarm Trigger will contain the hardware device automatically in its properties.**

### 3.1.1. Alarm Set Trigger

Alarm set triggers provide faster and more efficient Trigger creation. Alarm set triggers rely on previously created Alarm Sets to be used in the Trigger configuration. Certain scenarios can make Trigger creation very tedious and by using Alarm Set triggers, the workload is reduced and a streamline path for future manipulation is created.

For example, a Trigger composed of 15 alarms for a single OV input requires many devices in separate Triggers. Figure 3-6 shows a Trigger configured with many different alarms from a single device.

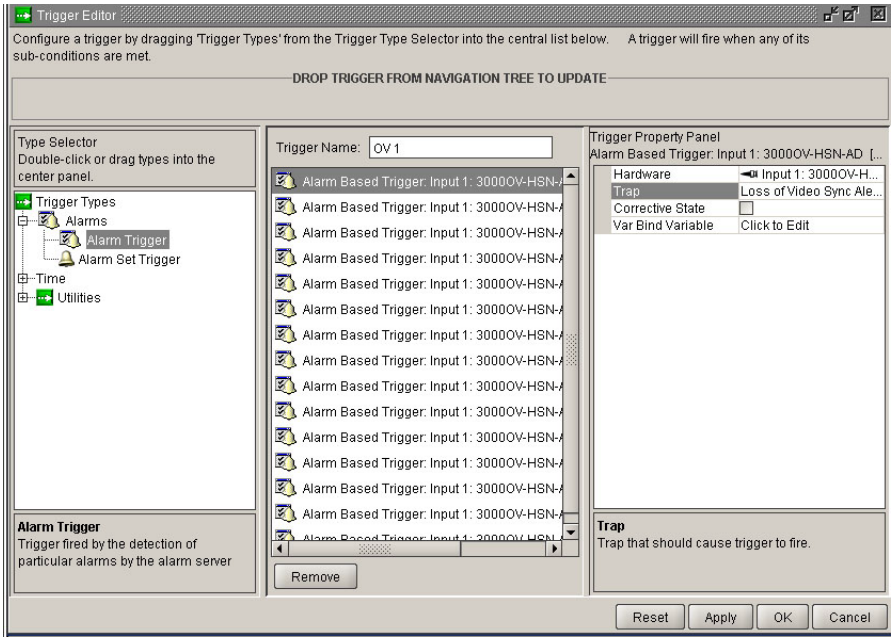



**Figure 3-6: Alarm Trigger**

Simply, an Alarm Set  is a template of Alarms. Traditionally Alarm Sets are created to perform Alarm Filtering for Service severities. These same alarm sets can be applied in a Trigger to enable the fire mechanism to activate for any alarm configured in the Alarm Set.

The first step to using *Alarm Set Triggers* is to create the Alarm Set. This is done by right clicking the Alarm Sets option and selecting *New -> Alarm Sets* from the pop up menu.

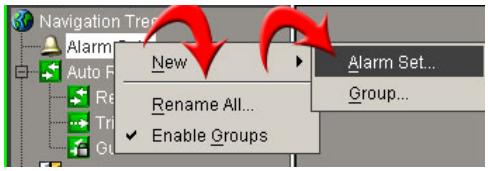



**Figure 3-7: Creating an Alarm Set**

**If the *Alarm Sets* option is not shown in the Navigation Tree, it can be enabled from the Tree Properties. Click on the 'Navigation Tree' icon than select 'Properties' from the application top drop down menu to expose the alarm sets option.**

Once the Alarm Set editor has opened, drag a device from the Navigation Tree to the area stating '*Drop Alarm set or Hardware from Navigation Tree here*' in the editor. The alarms for the device will be shown in the bottom list. Check off which alarms should be enabled for the Trigger. Create a name for the Alarm Set and click the 'Save' button.

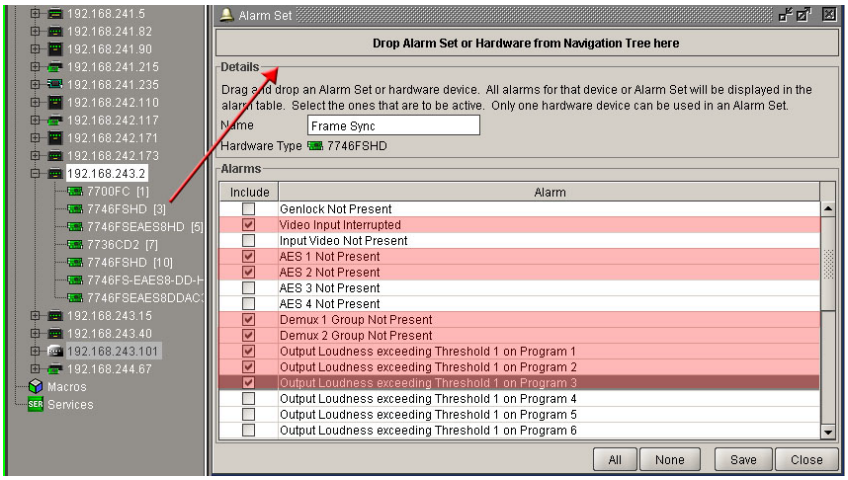



**Figure 3-8: Alarm Set Creation**

From the Trigger editor, drag the  Alarm Set Trigger component into the middle column. The right most column will be updated to show the relevant properties.

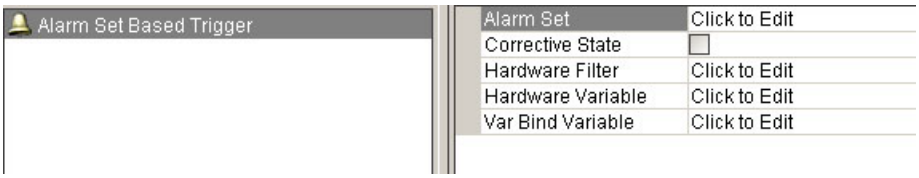



**Figure 3-9: Alarm Set Trigger Properties**

Alarm Set Triggers have five property values for configuration, as listed below.

- **Alarm Set:** This property is used to select the saved Alarm Set created. Click the *'Click to Edit'* text to expose any created Alarm Sets.

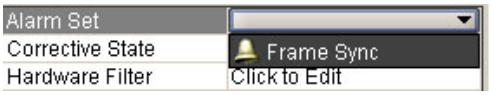



**Figure 3-10: Selecting a Alarm Set**

- **Corrective State:** The Corrective State parameter is used to enable the correction alarms for the alarm selected in the *Alarm Set*. For example, in Loss of Video, selecting the corrective state for that alarm would indicate that the alarm is sent when the video has regained connection to the device. This parameter is not mandatory to configure.

- **Hardware Filter:** This parameter is used to select which device(s) should contribute to the Alarm Set alarms. Multiple devices can be combined if needed. Most often only a single device is selected. Click the '*Click to Edit*' text to expose the "**…**" button. To choose a device from the tree, click the "**…**" button to open the Hardware Selector dialog. To select multiple devices, use the CTRL and SHIFT keys appropriately.

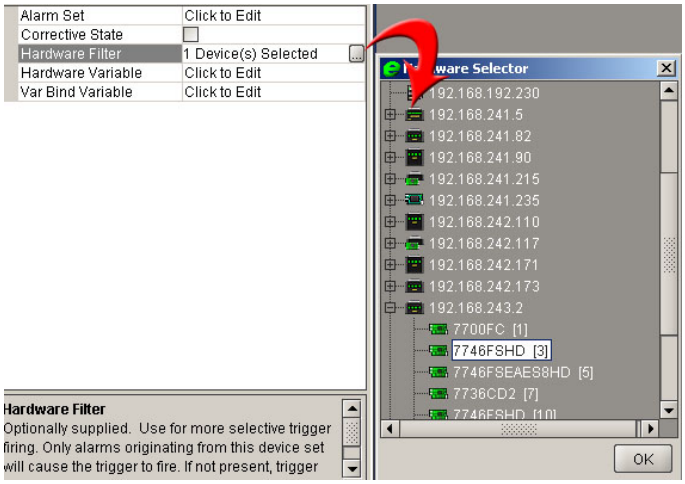



**Figure 3-11: Hardware Filter Selection**

- **Hardware Variable:** This parameter is used to assign the alarming device ID information to a hardware variable when the trigger is fired. The scope of the variable set is limited to the execution context of the response(s) initiated by this trigger. This parameter is not mandatory to configure.

- **Var Bind Variable:** This parameter specifies the index variable used to store the alarm's variable binding data for later processing and analysis. This parameter is not mandatory to configure.

**The same Alarm Set that is created can belong to multiple Alarm Set Triggers. Having a template like Alarm Sets is the benefit of Alarm Set Triggers.**

**It is possible to drag the Alarm Set from the Navigation Tree directly into the Trigger editor component list. The Alarm Set selector property will be automatically filled out.**

### 3.1.2. Scheduled Timed Triggers

Timed triggers have a different fire mechanism than alarm-based triggers. They rely on a calendar-based configuration to control their fire mechanisms. These types of triggers can leverage strategies where the system does not always have to react based on faults.

The following items describe the Timed based triggers that are available:

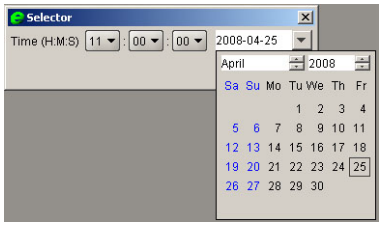
**Interval Trigger:** This type of Timed based trigger gets fired after a set period of time elapses. The Interval Trigger has three types of configurable properties, as listed below:

- **Interval:** The period of time between firing the Trigger
- **Unit:** Unit of measurement that the interval is expressed in. This can be millisecond, seconds, minutes, hours and days.
- **Delay:** This is a period of time (in ms) that should elapse before the initial firing of the Interval Trigger.

**One Shot** This type of Timed based trigger gets fired at a specific date and time. One Shot

**Trigger:** indicates that it is only fired once without it reoccurring. The One Shot Trigger has two types of configurable properties.

- **Date:** The Date/Time at which this trigger should be fired. The image below shows the calendar configuration.



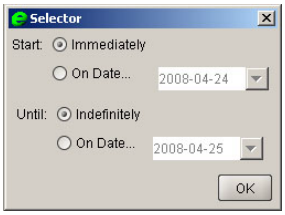

- **Advance Warning:** This parameter is used to warn clients of the trigger minutes prior to it's scheduled execution.

**Daily Trigger:** This type of Timed based trigger gets fired on a daily basis at a specific time. The Daily Trigger has four configurable properties, which are listed below.

- **Days:** The parameter specifies the day separations. For example, the day separations could be every day (1) or every two days (2).

- **Time of Day:** This parameter specifies the time of day it will fire.


- **Range:** This parameter specifies the time period over which this trigger is active.



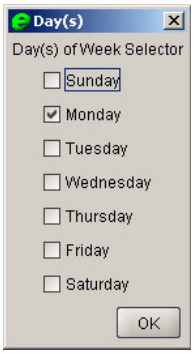

- **Advance Warning:** This parameter is used to warn clients of the trigger minutes prior to it's scheduled execution.

**Weekly Trigger:** This type of Timed based trigger gets fired on a weekly basis. The Weekly Trigger has five configurable properties, as listed below:

- **Weeks:** The parameter specifies the week separations. For example, the separations could be every week (1) or every two weeks (2).
- **Time of Day:** This parameter specifies the time of day the trigger will fire.
- **Range:** This parameter specifies the time period over which this trigger is

active.

- **Advance Warning:** This parameter is used to warn clients of the trigger minutes prior to it's scheduled execution.
- **Day(s) to Fire On:** This parameter specifies which day(s) of the week to fire on.




**Monthly Trigger:** This type of Timed based trigger gets fired on a monthly basis. The Monthly Trigger has five configurable properties as listed below:

- **Months:** The parameter specifies the week separations. For example, the separations could be every month (1) or every two months (2).
- **Time of Day:** This parameter specifies the time of day the trigger will fire.
- **Range:** This parameter specifies the time period over which this trigger is active.
- **Advance Warning:** This parameter is used to warn clients of the trigger minutes prior to it's scheduled execution.
- **Day of Month:** This parameter specifies which day of the month the trigger will fire on. Must be within the range of 1 to 31.

**It is possible to mix and match Timed Based triggers with Alarm Based Triggers.**

## 3.2. COMPOSED TRIGGERS

Composed Triggers are a utility feature that allows the editor to combine multiple created triggers into a single trigger. To use the Composed Triggers component, drag the Composed Trigger from the left column into the middle column. Only one property exists for the Composed Trigger. Click the property value to expose the "**...**" button. Select the "**...**" button and the Trigger Select dialog will appear.

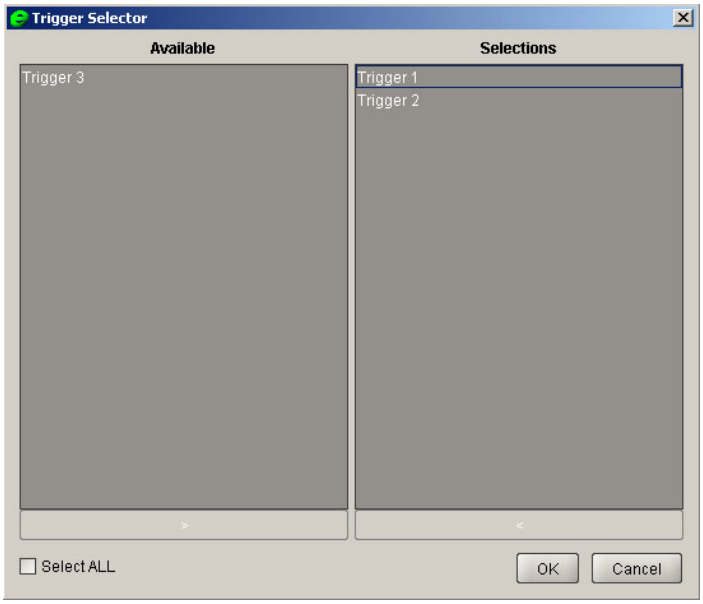



**Figure 3-12: Trigger Selector**

The dialog works by moving the available triggers from the left column to the right column by clicking the ">" button at the bottom. Once all the triggers are selected, click the 'OK' button to save the changes. The Composition component should resemble Figure 3-13.

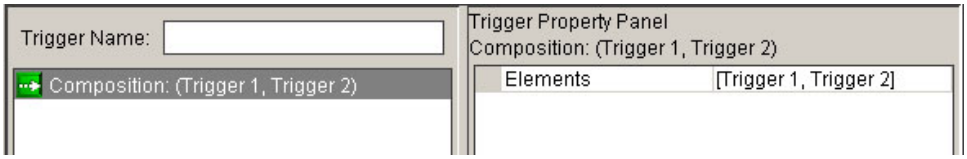



**Figure 3-13: Composition Trigger**

**It is possible to drag the trigger from the Navigation Tree and drop it into the editor component list for quicker configuration.**

### 3.2.1. Editing Triggers

To modify a Trigger, right click the desired trigger from the Navigation Tree and choose 'Edit'. The Editor will open and the trigger contents will be displayed. It is possible to drag and drop a trigger into the editor top panel to show the contents of the trigger. Figure 3-14 highlights the droppable area for the Trigger Editor so it will update automatically.

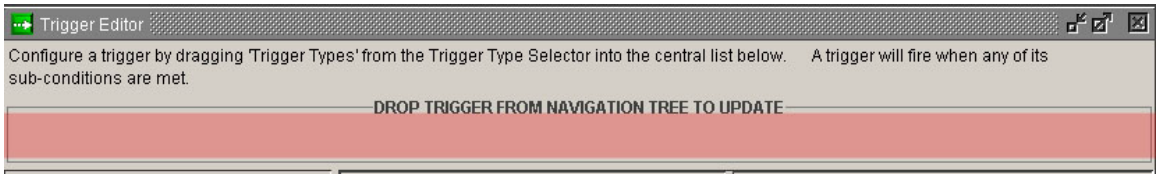



**Figure 3-14: Drop Area for Triggers to Update the Editor**

### 3.2.2. Arming and Un-Arming Triggers

Triggers can be disabled and enabled by right clicking the desired trigger and toggling the armed state. This is the quickest and easiest method to temporarily disable a trigger from activating. When triggers are not armed, they are represented in the Navigation Tree with a gray (unverified) icon.

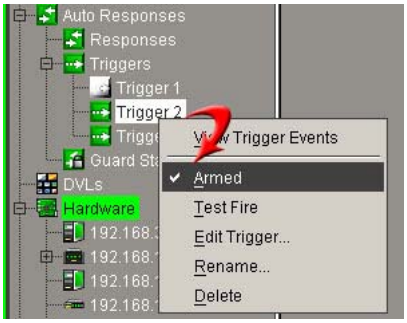



**Figure 3-15: Controlling the Armed State for a Trigger**

# 4. GUARD STATES

*Guard States* are an optional but very important entity to any automated systems in VistaLINK® *Auto Response*. Guard State creation is aided through a standard GUI dialog. The below topics explain the different types of guard states and how they can be used and managed.

## 4.1. GUARD STATE EDITOR

The Guard State editor is similar to the Trigger editor with respect to the configuration columns. The left most column enables the user to select the Guard type that is to be used. The middle column displays the contents of the Guard. The right most column acts as a properties panel, which displays how each Guard component is configured.

To create a new Guard, right click the 'Guard State' option from the Navigation Tree and select the 'New' option, then select 'GuardState…'.

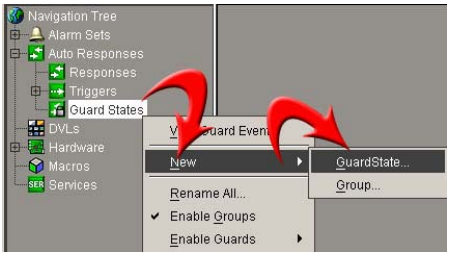



**Figure 4-1: Creating a New Guard State**

Figure 4-2 displays the Guard State Editor columns and control selection.

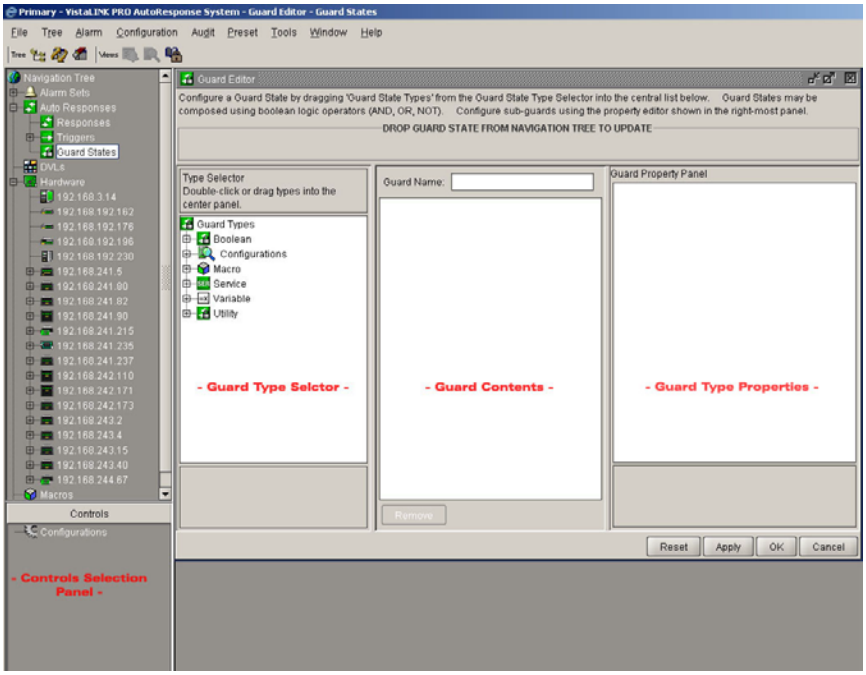



**Figure 4-2: Guard State Editor**

The *control selection panel* provides an interface for navigating through the parameters of various devices, enabling the user to select the parameters that need to be added to the Guard elements.

## 4.2. CONFIGURATION (INFORMAL)

A *Configuration Informal Guard* is the most common guard state that can be created. This type of Guard state matches a control parameter value to create a *Pass* or *Fail* evaluation. The below steps highlight how to create a *Configuration Informal Guard State*.

1. To create a *Configuration Informal*, select the *Configuration (Informal)* component from the *Type Selector* panel and 'drag and drop' it into the middle column. Figure 4-3 displays an item in the Guard Contents Column.

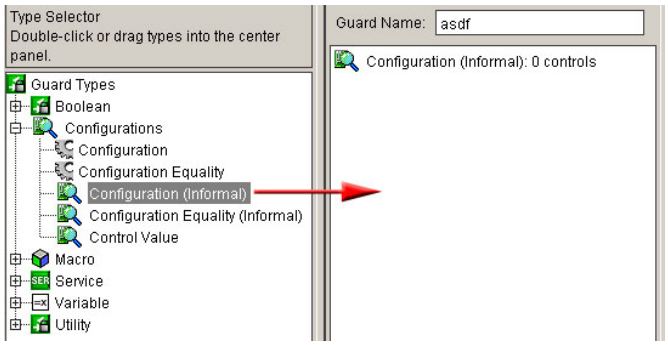



**Figure 4-3: Using the Configuration Informal Component**

2. The next step is finding a control parameter that the user can use to build the *Configuration Informal*. Select the desired device from the Navigation Tree and drag and drop it into the below Controls Selection. Figure 4-4 displays the *Control Selection* populated with a hardware device.

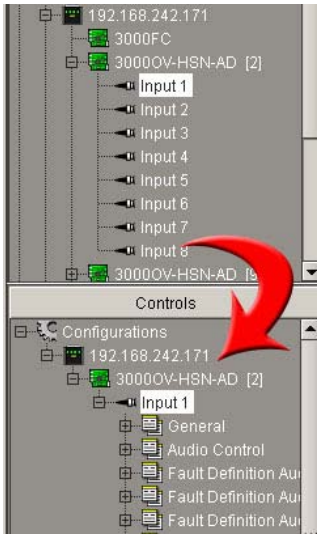



**Figure 4-4: Adding a Device to the Control Selections**

3. Select the desired control from the *Control Selection* panel and drag and drop the control into the Guard Editor (right most column). This would be the property area for the selected item in the Guard component list. Figure 4-5 illustrates where the control parameter should be placed.

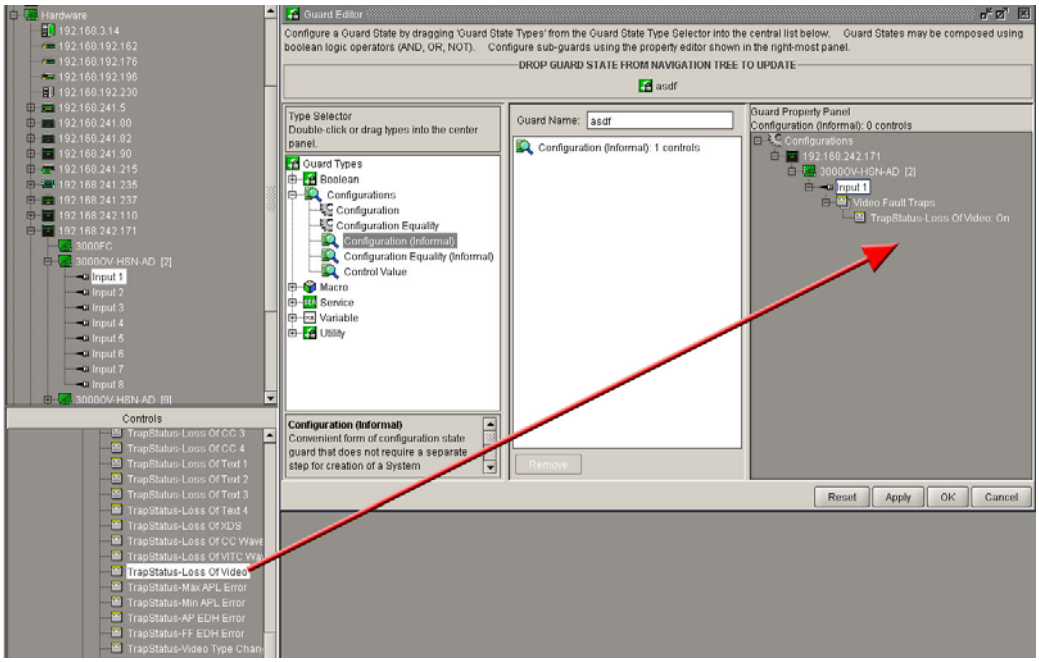



**Figure 4-5: Dropping Controls from Control Selection to Guard Editor**

**It is only possible to have one hardware device in the Controls Selection pane at a time. Adding additional devices would automatically force the current device out of the pane.**

4. The next step is to configure the matching condition for the parameter. Double click the parameter to expose the small editor. During the 'matching' process, the user must keep in mind the condition that the value must be set to in order for the Guard State to *pass*. Figure 4-6 identifies the appearance of the Trap Status parameter.


**Figure 4-6: Matching Condition Editor**

Double click the parameter to change the value. For this instance, double clicking the red box will change it to a green box. The editor will show the correct control depending on the parameter being used.

Select the Combo box to change the operator matching parameter. The different values are:

- Equals
- Not Equals
- Greater Than
- Greater Than or Equals
- Less Than
- Less Than or Equals

5. The last step is to save the Guard State by clicking either the 'Apply' button or the 'OK' button. Ensure a descriptive name is inserted into the **Guard Name** field. An example of the guard name is identified in Figure 4-7.


**Figure 4-7: Creating a Guard Name**

**For more efficient use, you can eliminate Step 1 by adding the parameter from the Controls Selection pane directly into the guard editor (Middle column). The application will prompt the type of component that the parameter should be made out of.**

Additional parameters can be combined into a Configuration Informal component. When doing this, all parameters have to pass the matching conditions for the Guard to *Pass.* This creates an AND'd condition for everything in the Configuration Informal component. If one does not match properly to the matching condition, the whole card will *Fail*. Figure 4-8 displays multiple parameters from different devices combined together.

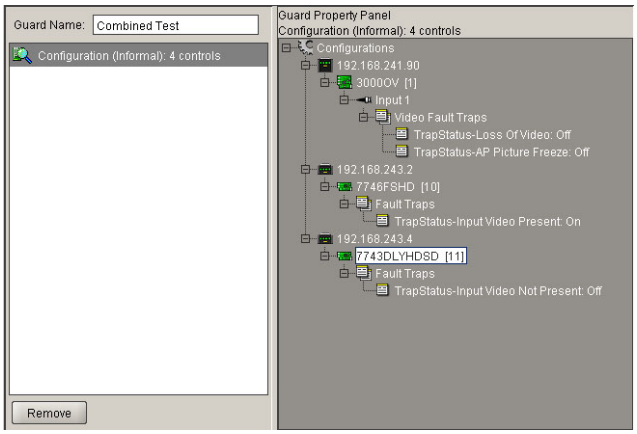



**Figure 4-8: Multiple Parameter Conditions**

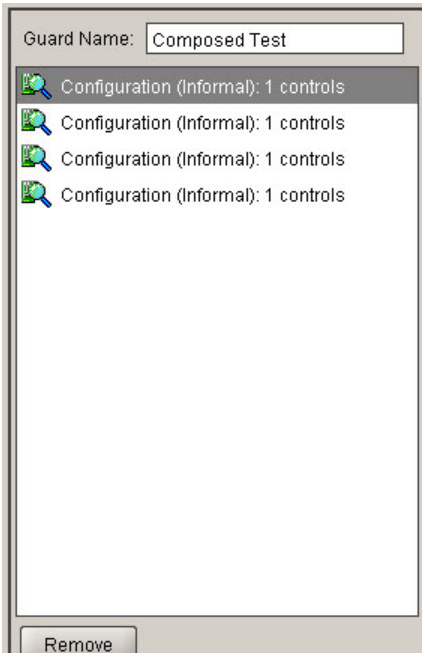



**Figure 4-9: Multiple Components Test**

Another method that will **AND** multiple parameters together is to use multiple *Configuration Informal* components. The benefit of this approach is granular organization when making large and complex guard components. It is possible to mix and match both methods together. Figure 4-9 displays the appearance of the multiple components.

### 4.2.1. Configuration Equality (Informal)

Configuration Equality guards are a special type component that has a higher level of matching. The goal for this Guard type is to match all values in the guard to be equal to each other. This means that if all the control parameters have the same value, then the guard will *Pass*. If one parameter has a different value than any other parameter, the entire guard state evaluation will *Fail*. Popular strategies for this are to compare if signal path A is in the same state as signal path B.

Use the Configuration Equality (Informal) outlined in the below steps.

1. To create a Configuration Equality Informal, select the Configuration Equality (Informal) component and drag and drop it into the middle column. Figure 4-10 displays the Guard Contents column with an item in it.

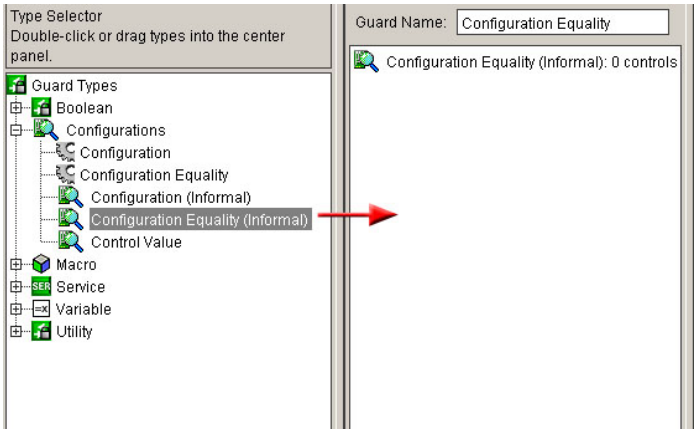



**Figure 4-10: Adding the Configuration Equality (Informal)**

2. The next step is to find a control parameter to build the *Configuration Equality Informal* with. Select the desired device from the Navigation Tree and drag and drop it into the below *Controls Selection*. Figure 4-11 displays the *Control Selection* populated with a hardware device.

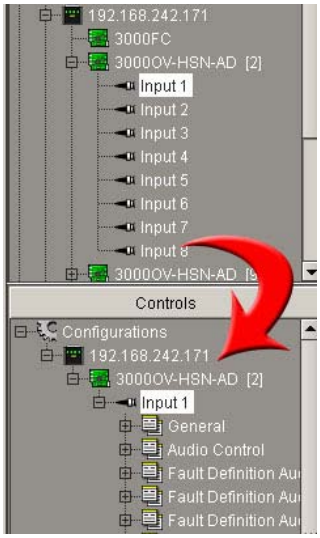



**Figure 4-11: Adding a Device to the Control Selections**

3. Select the desired control from the *Control Selection* panel and drag and drop the control into the Guard Editor right most column. This would be the property area for the selected item in the Guard component list. Figure 4-12 illustrates where the control parameter should be placed.

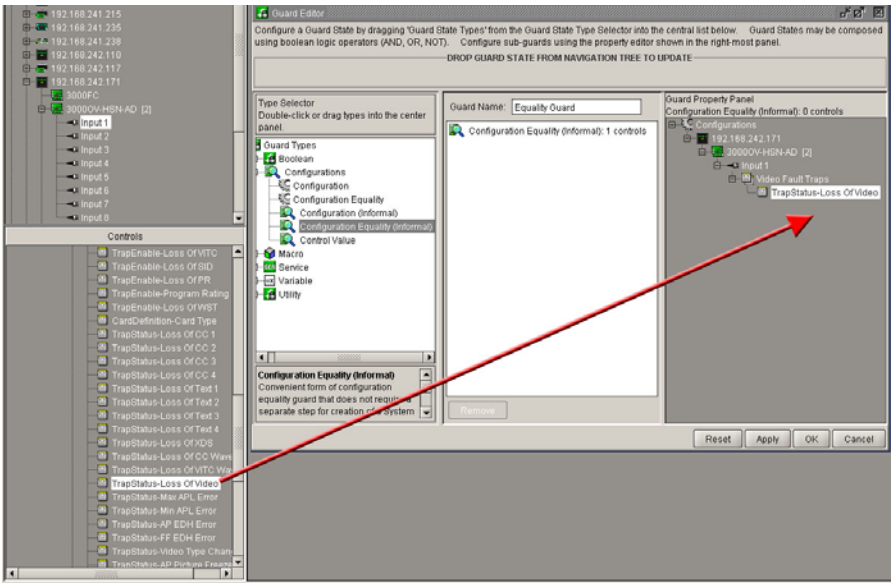



**Figure 4-12: Dropping Controls from Control Selection to Guard Editor**

4. Since the *Configuration Equality* guards work best with more than one control parameter, a second parameter can be added to the *Configuration Equality* component. Figure 4-13 displays how the component is composed of two MVP inputs matching the same parameter.

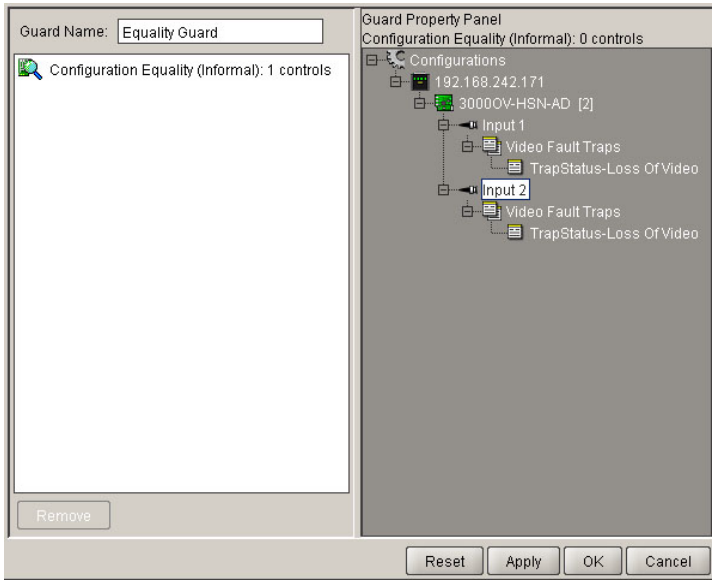



**Figure 4-13: Multiple Control Parameters for Configuration Equality**

5. The last step is to save the Guard State by clicking either the 'Apply' button or the 'OK' button. Ensure a descriptive name is inserted into the **Guard Name**. An example of the guard name is identified in Figure 4-14.


**Figure 4-14: Creating a Guard Name**

**For more efficient use, you can eliminate Step 1 by adding the parameter from the Controls Selection pane directly into the guard editor (Middle column). The application will prompt the type of component the parameter should be made out of.**

It is possible to build a Guard State with multiple Equality components. When using multiple components, the guard state has multiple compare groups. The control parameters in one Equality component does not contribute to any matching in another component.

With this method, it is possible to build up a single Guard State that can perform the following function:

(*Video Missing A* = Video Missing B) and (*Video Black A* = Video Black B) and (*Video Freeze A* = Video Freeze B)

If both channels have a simultaneous black signal, it will prevent the guard from Failing when being evaluated. Figure 4-15 identifies a Guard State built with multiple Configuration Equality components.


**Figure 4-15: Multiple Component Equality**

## 4.3. CONFIGURATION AND CONFIGURATION EQUALITY (THE NON INFORMAL METHODS)

Non-informal methods are components that can re-use existing batch configurations that have been made. The matching only has a hidden EQ operator. These components are seldom used but can provide the benefit of re-using existing setup.

To use either the *Configuration* or *Configuration Equality* components, drag either of them from the component selector tree to the middle column in the Guard State editor.

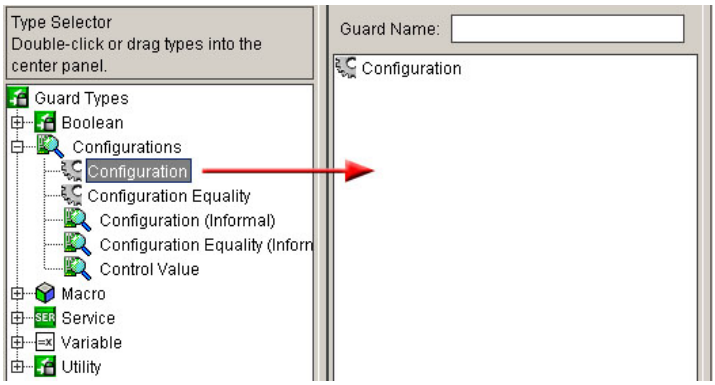



**Figure 4-16: Adding Configuration Component to the Component list**

Once the item is added it is possible to select a pre-made batch configuration for its property.

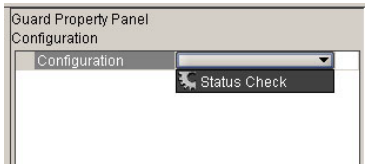



**Figure 4-17: Properties for the Configuration Component**

## 4.4. CONTROL VALUE

The *Control Value* component provides the same functionality as the *Configuration Informal* component model but at a more granular approach. The purpose of the Control Value is to allow for variables to key in certain values for different types of properties. A variable can define a hardware device or even the value itself. This type of strategy helps make the Guard State property values dynamic. Something external would be required to drive the variable to have the variable contents to be compatible.

The use of the Control Value component is described in the below steps.

1. To create a Control Value, select the Control Value component and drag and drop it into the middle column. Figure 4-18 displays the Guard Contents column containing an item.

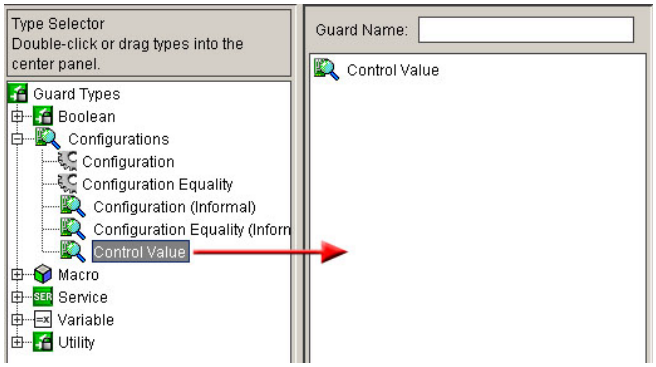



**Figure 4-18: Adding Control Value Component**

2. The properties column automatically updates to show the relevant properties. To specify a hardware device manually, click the 'Click to Edit' text to expose the "…" button. Click the "…" button to open the hardware selector.

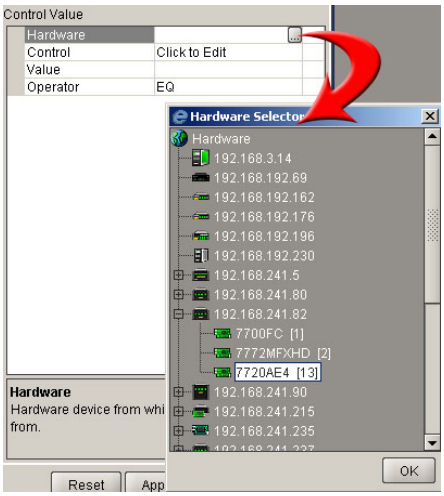



**Figure 4-19: Hardware Selection Dialog**

To use a Hardware Variable instead of a static hardware assignment, click the box on the very right side to enable the Variable mode. Once enabled, it will be possible to select a Hardware Variable for use. Only Hardware Variables that have been created will be shown.

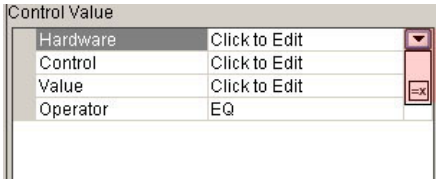



**Figure 4-20: Enabling Variables**

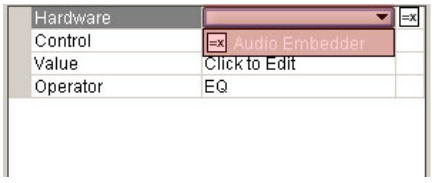



**Figure 4-21: Selecting a Hardware Variable**

3. To specify a control parameter for a hardware device, manually click the 'Click to Edit' text to expose the "…" button. Click the "…" button to open the control selector.

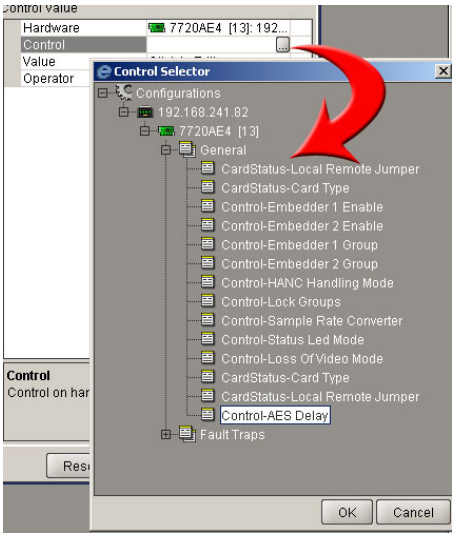



**Figure 4-22: "…" button**

**The Control Selector will only open if a hardware device is specified from either a constant or a variable setting.**

The Control property does not support a variable at this time of writing.

4. To specify a constant Value for the hardware device control. Click the 'Click to Edit' text to expose the "…" button. Click the "…" button to open the Parameter Value editor.

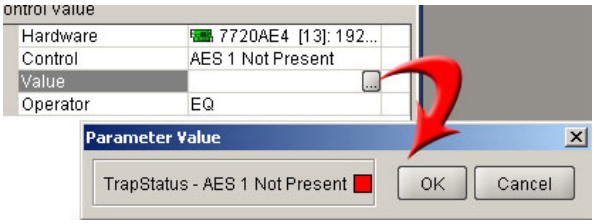



**Figure 4-23: Parameter Value Editor**

To use the Variable options, click the right column beside the value and select a variable type. There are two types of variables that can be used for the Value property.

- **Variable:** When using this option, any previously created Text Variables will be shown from the drop down menu.

- **Expression Inline Variable:** When using this option, the variable can be built from a small Variable editor by clicking the "…" button.

5. The Operator property allows you to specify an operator to be used for the *value match* condition. Click the 'EQ' text to expose the drop down for other options.

**The Control Value component is only valuable if the strategy involves using a Variable to control the hardware or matching value properties.**

## 4.5. IS MACRO CYCLING

This type of Guard State Component is used to test whether a Cycling Macro is running or not. The Guard State evaluates a Pass if the Cycling Macro is currently running. To use the *Is Macro Cycling* component, select it and drop it into the middle column of the Guard State editor. The component only has one property to it, which allows the selection of the Cycling Macro to be tested. It is possible to use a variable instead to specify the Cycling Macro. Figure 4-24 shows the property adjustment for the component.

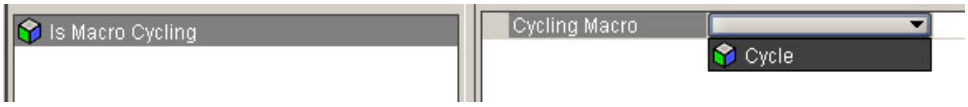



**Figure 4-24: Is Macro Cycling Properties**

## 4.6. VARIABLE VALUE AND INDEXED VARIABLE VALUE

These two Guard State components allow the Guard State to evaluate internal variables that are built into the system. Each has relatively the same property values to it. They are listed below:

1. Select the variable to test against. A constant value or a variable value setting can configure this.
2. The value of the variable for a correct match to occur. This can be a constant value or a variable value. The Σ Expression Builder is available to define the value from the Guard State editor.
3. The type of operator that the test will use (EQ, NEQ, LT, LTE, GT, GTE)
4. The Index value for the row to be used when using an Indexed Variable. This value can be constant or be supplied by a variable. Either from a pre-built variable or from the Σ Expression Builder.

## 4.7. IS HARDWARE IN SERVICE

The SER *Is Hardware In Service* component is a method of satisfying the Guard State evaluation based on if a particular device is in service or not. If the hardware is encapsulated into a Service then the Guard State evaluation will *Pass*. To use the component, drag and drop it from the left column into the middle column. Only two properties exist for the *Hardware In Service*.

- **Service:** This property is used to select from which service to evaluate. A constant value can be specified or a Variable can drive it.
- **Hardware:** The device to evaluate for the above service. Click the 'Click to Edit' text to expose the "…" button. Select the "…" button to expose the Hardware Selector. It is possible to specify a Variable of the property for dynamic use.

## 4.8. BOOLEAN COMPOSED GUARDS

The Boolean components are used to compose multiple Guard States into a single Guard State. This allows for re-using existing guards that have been created for other profiles. It helps to reduce duplicate work and also make sure Guard States are less complex and easier to troubleshoot.

The three types of Boolean components are listed below:

- **AND:** This component works by doing a logical AND for all the Guard States added to this component. If any of the Guard States evaluate as a *Fail*, then the whole Guard State will *Fail*.
- **OR:** This component works by doing a logical OR for all the Guard States added to this component. This component is useful since it is not possible to do any logical OR's when using multiple components or multiple control parameters from within a single Guard State. If any of the Guard States in this component evaluate as a Pass, than the whole Guard State will *Pass*.
- **NOT:** This type of component reverses the evaluation of a AND guard. This Guard Sate will evaluate a Pass when all of the composed guard states evaluate a Fail.

To use any of the above Boolean components, add the component from the left column to the middle column.

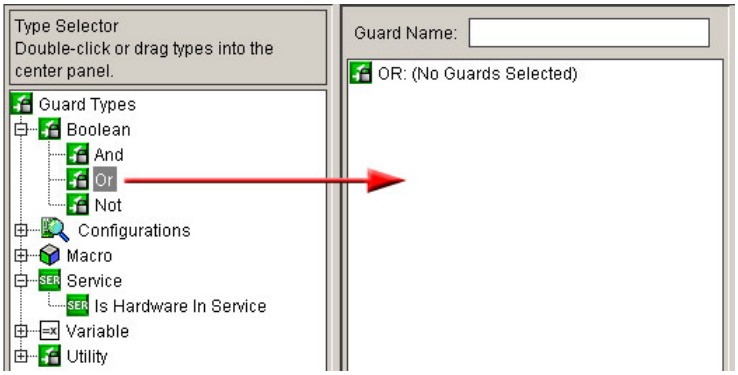



**Figure 4-25: Adding OR Component to the Component List**

Only one property exists for the Boolean type component that is to select which Guard States should contribute to the component. Click "…" button from the property field to expose the Guard Selector. Figure 4-26 displays the Guard Selector. The left column shows the available Guard States and the right column lists the chosen Guard States. Use the bottom ">" arrow button to push items from one list to the other list. Click 'OK' to save the changes.

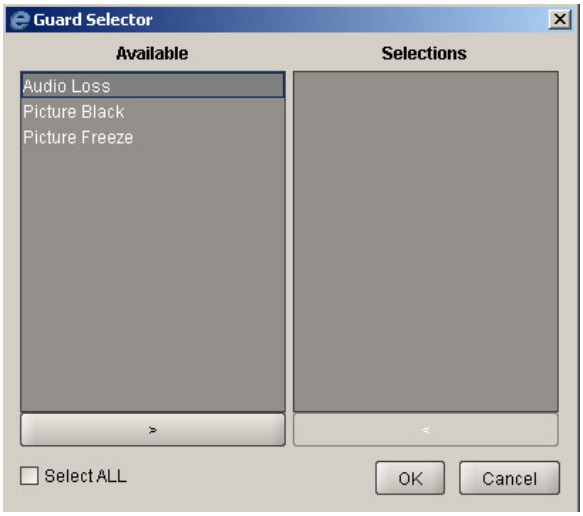



**Figure 4-26: Guard State Selector**

**It is possible to drag and drop a previously made Guard State directly into the editor (middle column). The confirmation popup will appear requesting the type of compose component it should be.**

## 4.9. EDITING GUARD STATES

The Guard State editor has a droppable area at the top of the dialog that previously made Guard States can be dropped onto. By doing this, the editor is updated automatically to show the contents of the Guard State. Figure 4-27 highlights the droppable area to add a Guard State too.

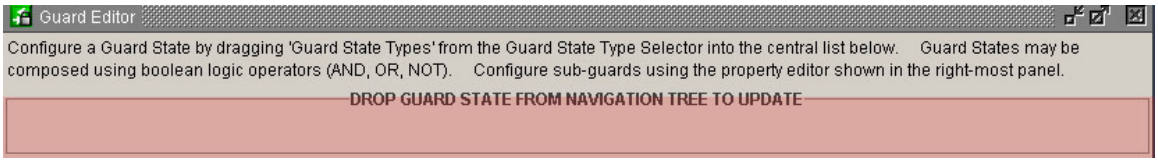



**Figure 4-27: Editing Guard States**

**It is also possible to right click a Guard State and choose the 'Edit' option. The editor will open automatically showing the contents of the Guard.**

## 4.10. TESTING GUARD STATES

It is simple to test a Guard State to ensure that the evaluation is operating as expected. Instead of building an actual Auto Response profile to test with, each Guard State has an 'Evaluate Guard' function in the right mouse click menu. To access the menu, select and right click a Guard State, then select the 'Evaluate Guard' option. A popup message will appear stating the results of the evaluation.

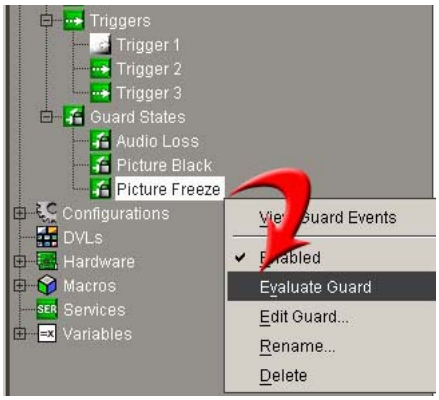



**Figure 4-28: Evaluating a Guard State Manually**

# 5. RESPONSES

## 5.1. RESPONSE EDITOR

*Auto Response* relies on the *Response Editor*  to create the automated profiles. The *Response Editor* is a GUI that links the **Triggers**, **Guard States** and **Payloads** together. To access the *Response Editor*, right click 'Responses', press 'New' and then select '***Auto Response...'***. Figure 5-1 illustrates the selections.

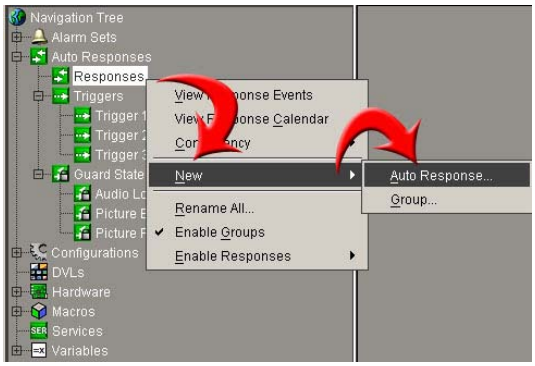



**Figure 5-1: Creating a New Auto Response**

The *Response Editor* enables the user to drag and drop the *Navigation Tree* elements into the GUI. Specific areas in the *Response Editor* only support certain types of elements. This means that it is not possible to drag a *Guard State* into a Trigger area.

Figure 5-2 displays the *Response Editor* and the droppable areas.

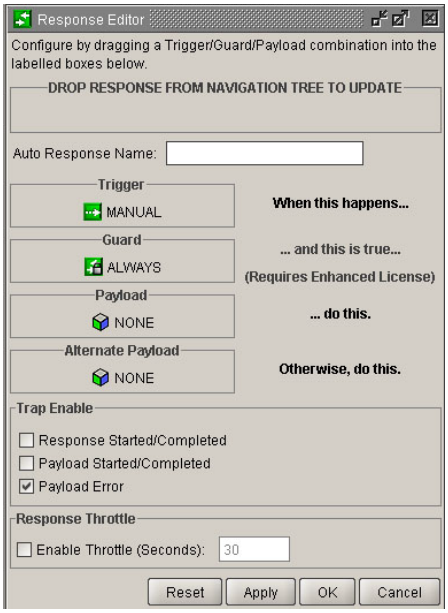



**Figure 5-2: Response Editor Full**

The main *Response Editor* categories are divided into the following execution stages:


- **Triggers:** Triggers from the Navigation Tree can be dropped directly into the *Response Editor*.


- **Guard States:** Guard States from the Navigation Tree can be dropped directly into the *Response Editor*. This entity is optional and if a Guard State is omitted, the Guard State evaluation is *True* by default.

**If the *Response Editor* has a message beside the Guard States saying "Requires Enhanced License", it is not possible to use Guard States in the system. For Guard States to work with Response profiles, Auto Response Full is required in the VistaLINK® License.**


- **Payload:** Macro's from the Navigation Tree can be dropped into this field. This is the macro that should be fired if the Guard State evaluation is *True*. This entity is optional.


- **Alternate Payload:** Macro's from the Navigation Tree can be dropped into this field. This is the macro that should be fired if the Guard State evaluation is *False.* This entity is optional.

Once all the execution stages are configured, the Response requires a custom name for it to save. Click either the 'Apply' button or the 'OK' button to save the configuration.


**Figure 5-3: Auto Response Name Field**

## 5.2. TRAP ENABLE

The **Trap Enable** parameters in the *Response Editor* Enable/Disable the specialized Auto Response Traps. These alarms aid in troubleshooting and to track/alert about *Auto Response* profiles being executed. Three alarms exist for every profile

- **Response Started/Completed Alarm:** This parameter controls whether a notification is created when the profile starts and stops.
- **Payload Started/Completed Alarm:** This parameter is used to notify when the Auto Response profile executes the Macro. It traps when the Macro starts and also when the Macro stops.
- **Payload Error:** This is a notification that is sent when the contents of the Macro cannot be executed. This parameter is useful to track failures when running configurations.

## 5.3. RESPONSE THROTTLE

The Response Throttle can be used for alarm based *Auto Response* profiles. Often an Alarm trigger can be configured to fire based on many alarms. Some alarms can occur because of a major alarm (like Loss of Video). When using the throttling option, it is possible not to delay the execution of the *Auto Response* profile for a certain duration after it executes.

If a 'Loss of Video' alarm occurred, the alarm would trigger consecutive Picture Freeze, Picture Black and Loss of Audio alarms. This would cause an Auto Response profile to execute more than one time, which may be undesirable. The throttle mechanism would prevent the secondary alarms from triggering the profile to execute more than once.

**By default, Auto Response profiles are executed in parallel threads. This indicates that the system does not wait until one response finishes to execute another response. If a single Auto Response profile is triggered multiple times at once, there will be multiple instances of the Auto Response running.**

When Response Throttles are in use, the system discards any calls to execute the Auto Response profile until the timer is finished. This functionality is often not desirable because the secondary wave of the alarms could mean a successful Guard State evaluation. Due to the operation of the throttling, Auto Response has a secondary management mechanism that serializes the Auto Response executions. This feature is called Response Locks, which is explained in the following section.

## 5.4. CONCURRENCY – AUTO RESPONSE LOCKS

The *Auto Response Locks* functions in the opposite manner of Response Throttles. Rather than discard execution calls temporarily for certain time duration, Response Locks prevent the multi-threaded nature on how response profiles are executed. In a scenario where Response Locks are enabled for a profile, if the profile is called for execution multiple times in one short time frame the response profile will execute efficiently rather than all at once. This creates stability and prevents incorrect Guard State evaluation in certain scenarios.

To use Auto Response Locks, right click the 'Responses' option from the Navigation Tree and select 'Concurrency' and then click the 'Manage Locks' option.

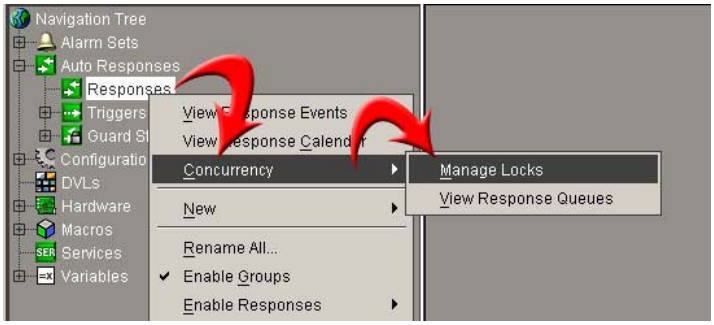



**Figure 5-4: Accessing the Response Locks**

Once the Response Lock Editor opens, enter a Lock Name for the virtual group and click the 'Create' button.


**Figure 5-5: Creating a Locking Group**

After clicking the 'Create' button, double click the Locks icon to expose the newly created locking group. To add a response to the group, drag it from the Navigation Tree directly into the Response Lock group. Figure 5-6 illustrates the auto response locks.

**If the *Auto Response* profiles are daisy chained together, only the first *Auto Response* profile needs to be in a lock. The first profile does not terminate completely until the last profile has finish executing.**

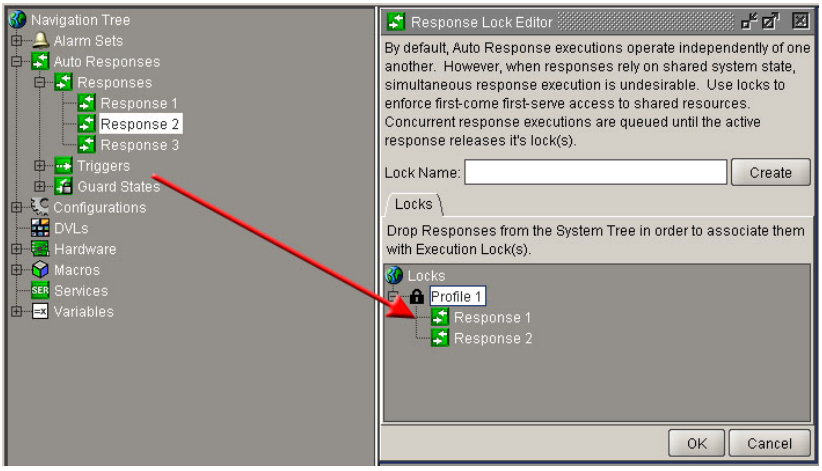



**Figure 5-6: Auto Response Locks**

**Multiple Response Lock groups can be created and utilized independently of each other.**

## 5.5. AUTO RESPONSE CALENDAR

The *Auto Response* Calendar is a tool used to investigate which *Auto Response* profiles execute on which days. The system automatically determines this information based on the type of Triggers used in the *Auto Response* profiles. To open the calendar, right click the 'Responses' option and choose 'View Response Calendar'. Figure 5-7 displays the calendar.


**Figure 5-7: Response Calendar**

## 5.6. AUTO RESPONSE EVENT MONITORING

Auto Response has an event viewer that can display current and past events. The event viewer can display events for a certain element or group of elements. The event viewer is useful for monitoring and troubleshooting system developments and designs. Right click any Auto Response element and select the 'View Events' option to execute the event viewer. Figure 5-8 displays the event viewer.

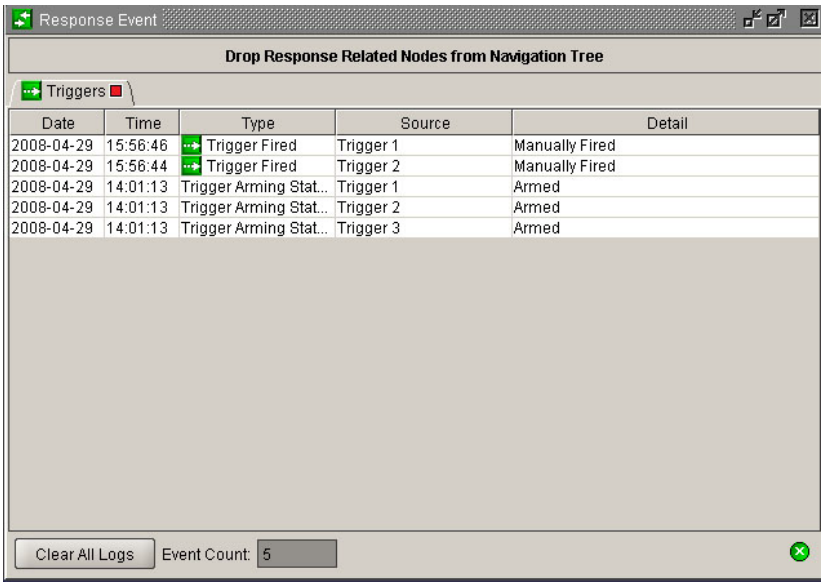



**Figure 5-8: Event Viewer**

The event viewer lists the events from the most recent to the oldest, as well as time and date stamp information. The detail column provides a description of the event. The Type and Source parameters identify the type of trigger action and the Trigger source number. The event viewer enables a new *Auto Response* node to be dropped into the GUI. A tab will automatically be created for the logging of that element. Click the red box on the tab to close the tab.

Figure 5-9 highlights the droppable area, which is used for adding event views to the event viewer.

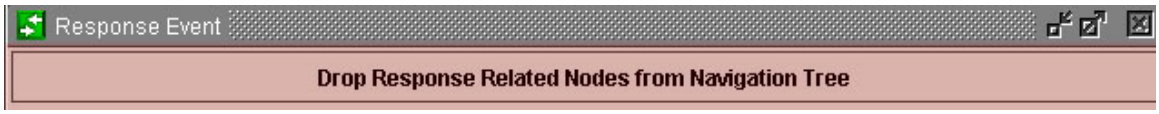



**Figure 5-9: Event Viewer Drop Area**

The right hand bottom corner button enables the suspension of new events from showing in the event viewer. If the button goes from a disabled state to an enabled state, past events that have been buffered will be present in the main log area.