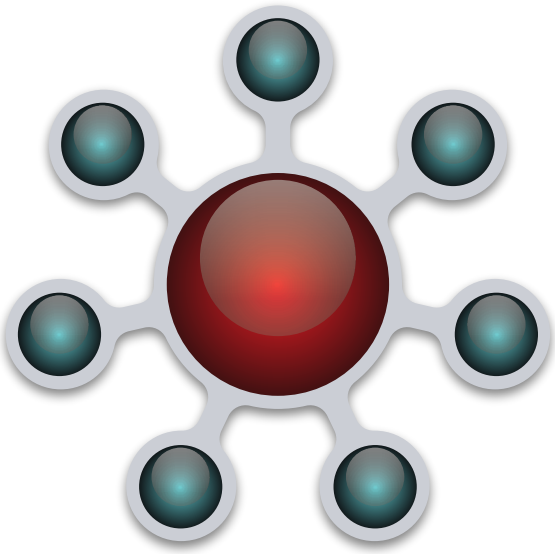
# STARDRAW CONTROL 2010

*ultimate control*



# HALOGEN

**Halogen Software**


Stardraw Control is a trademark of Stardraw.com Ltd.

# Table of Contents


**CHAPTER 1: Overview** ..... **1**

About This Document ..... 1

Using the HAL System Documentation ..... 1

Getting Started ..... 3

**CHAPTER 2: Introduction to Using External Control Systems with HAL** ..... **4**

Configuring External Controls in Halogen ..... 4

Connecting to a Halogen/HAL Control Server ..... 4

Communicating with a Halogen/HAL Control Server ..... 5

Using a Telnet Client to Test and Monitor the System ..... 5

**CHAPTER 3: Example HAL1x Configuration** ..... **6**

Creating the Example Configuration ..... 6

Configuration Summary ..... 15

**CHAPTER 4: Integrating Stardraw Control Systems with HAL** ..... **16**

Overview ..... 16

Checklist ..... 17

Getting Started ..... 17

Overview of Stardraw Control Application Development ..... 17

Importing the Example Stardraw Project ..... 17

Understanding the Example HAL Driver ..... 20

Part 1 - Communicating with the Halogen/HAL Control Server ..... 20

Part 2 - Connecting to the Stardraw Application ..... 22

The Complete Driver Application Interface ..... 24

Exploring the Stardraw Example Application ..... 25

Overview ..... 25

Toggles ..... 25

Commands ..... 26

Levels ..... 27

Selectors ..... 29

Initializing the Form on Application Start ..... 33
Events/Actions Report ..... 33
Running and Testing the Application ..... 34
Troubleshooting ..... 38
**APPENDIX A: HAL External Control Message Protocol ..... 40**
Details of external control messages ..... 40
All Controls ..... 41
Level Controls ..... 41
Toggle Controls ..... 43
Selector Controls ..... 45
Command Controls ..... 48
Communications Monitoring ..... 50
**APPENDIX B: Using PuTTY to Test External Control Systems ..... 51**
**Trademarks ..... 57**

# CHAPTER 1: Overview

## About This Document

This guide explains how to use a HAL DSP Processor with Stardraw control systems. It is divided into three major sections:

- "Introduction to Using External Control Systems with HAL" on page 4 (Chapter 2)
- "Example HAL1x Configuration" on page 6 (Chapter 3)
- "Integrating Stardraw Control Systems with HAL" on page 16 (Chapter 4)

Chapter 2 explains the general approach for configuring a HAL for use with any external control system and how to connect and communicate with the HAL Control Server. The second section, Chapter 3, walks you through the construction of an example Halogen configuration that includes one or two common uses for each type of external control. Chapter 4 shows you how to set up and program a Stardraw control system to work with a HAL that is running the example configuration.

The appendices provide helpful reference information about the HAL external control message protocol and how to use a telnet client to monitor and troubleshoot the operation of a control system at the message protocol level.

## Using the HAL System Documentation

A variety of documentation is available to help you get started with and use your HAL System:

### Halogen Help System

A comprehensive help system is installed with the Halogen software. It contains all the information you need to work with the system. There are several ways to access the Help System:

- Click the **Help** icon on the application toolbar: Clicking the icon itself opens the Help System. Clicking the down arrow displays a list of options including such things as access to the Rane website, checking for updates to the Halogen software, and sending an email to Rane.
- Click the Help icon that appears in the upper right corner of Halogen dialog boxes. Clicking this icon opens the Help topic related to the specific dialog box. From there you can access the entire Help System, if needed.
- Right-click on different elements in the user interface. A **Help** option appears in the context menu. Click this option to open the relevant Help topic. Pressing F1 when an area of the user interface has focus also displays its relevant Help topic.
- To search for information within the Help System, you can use the tabs on the Help Viewer's left pane to search the index (click the **Index** tab) or perform a full-text search (click the **Search** tab). You can also use the **Quick search** box in the Help System toolbar to search for text within the currently displayed topic:

### HAL System Design Guide

This guide is offered as a PDF file and contains a product overview, details about the HAL System's key features, and best practices for designing a HAL audio system. Note that the information in this guide is also available in the Halogen Help System. You can find this guide on the Rane website (http://rane.com/hal) as

well as on the product DVD.

### HAL System Installation Guide

This guide is offered as a PDF file and includes step-by-step instructions on installing the HAL hardware, loading a configuration, and testing the system. This information is also included in the Halogen Help System. You can find this guide on the Rane website (http://rane.com/hal) as well as on the product DVD.

### AMX Control Systems Guide

This guide, includes an introduction to using external control systems with HAL. It also discusses an example HAL1 configuration and how to set up an AMX controller and touch panel to communicate with a Halogen/HAL Control Server. In addition, an appendix is included with reference information on the HAL external control message protocol and how to use a telnet client to monitor and troubleshoot the operation of a control system at the message protocol level.

The guide is designed to be used in conjunction with the files found in the AMX Support Package. The contents of this support package include:

- AMXControlSystems_Guide.pdf - The AMX Guide pdf file
- ControlSystemSample.hal - Halogen configuration file intended for loading in Halogen or your HAL1x
- Rane_HAL.apw - NetLinx Studio program project file
- Main.axs - NetLinx program source file
- Rane_HAL_TP.TP4 - TPDesign4 touch panel project file

The AMX Support Package is installed with the Halogen software and can be accessed from the Windows Start Menu under Halogen -> Guides. If you want to access the files directly they are available on the product DVD or from the Halogen install directory:

- C:\Program Files (x86)\Rane Corporation\Halogen\Guides\Support Packages\AMX directory

You can also download the most up-to-date version of this support package from the Rane website (http://rane.com/hal).

### Crestron Control Systems Guide

This guide, includes an introduction to using external control systems with HAL. It also discusses an example HAL1x configuration and how to set up a Crestron controller and virtual touch panel to communicate with a Halogen/HAL Control Server. In addition, an appendix is included with reference information on the HAL external control message protocol and how to use a telnet client to monitor and troubleshoot the operation of a control system at the message protocol level.

The guide is designed to be used in conjunction with the files found in the Crestron Support Package. The contents of this support package include:

- CrestronControlSystems_Guide.pdf - The Crestron Guide pdf file
- ControlSystemSample.hal - Halogen configuration file intended for loading in Halogen or your HAL1x
- Rane_HAL.smw - SIMPL Windows program project file
- Rane_HAL_TP.vtp - VisionTools Pro-e touch panel project file
- Rane HAL Level Processor.usp - SIMPL+ user module source file
- Rane HAL Level Processor.ush - Compiled user module

The Crestron Support Package is installed with the Halogen software and can be accessed from the Windows Start Menu under Halogen -> Guides. If you want to access the files directly they are available on the product DVD or from the Halogen install directory:

- C:\Program Files (x86)\Rane Corporation\Halogen\Guides\Support Packages\Crestron directory

You can also download the most up-to-date version of this support package from the Rane website (http://rane.com/hal).

### Stardraw Control Systems Guide

This guide, includes an introduction to using external control systems with HAL. It also discusses an example HAL1 configuration and an example Stardraw Control application and driver that communicates with a Halogen/HAL Control Server. In addition, an appendix is included with reference information on the HAL external control message protocol and how to use a telnet client to monitor and troubleshoot the operation of a control system at the message protocol level.

The guide is designed to be used in conjunction with the files found in the Stardraw Control Support Package. The contents of this support package include:

- StardrawControlSystems_Guide.pdf - The Stardraw Control Guide pdf file
- ControlSystemSample.hal - Halogen configuration file intended for loading in Halogen or your HAL1x
- HAL1 Stardraw Example Project.s03 - Stardraw Control Project
- HAL1 Example Driver.cs - Stardraw HAL1 driver source code. This is part of the Stardraw Control Project but is included separately here for reference.

The Stardraw Control Support Package is installed with the Halogen software and can be accessed from the Windows Start Menu under Halogen -> Guides. If you want to access the files directly they are available on the product DVD or from the Halogen install directory:

- C:\Program Files (x86)\Rane Corporation\Halogen\Guides\Support Packages\Stardraw Control directory

You can also download the most up-to-date version of this support package from the Rane website (http://rane.com/hal).

## Getting Started

Read this document in sequence if you are completely new to learning about and using external control systems with HAL and the Halogen software. Chapters 2 and 3 appear in other control systems guides so you can skip these if you've already read them. The Stardraw example in chapter 4 refers to the example HAL1x configuration often, so you may want to familiarize yourself by reading chapter 3 before starting on chapter 4.

# CHAPTER 2: Introduction to Using External Control Systems with HAL

This section is a brief overview of the support that the HAL system provides for external controls and how your system can connect to and use these controls. For more information on this topic, please see the Halogen Online Help.

## Configuring External Controls in Halogen

Halogen provides the ability to configure controls for use with external control systems. These controls can be of any of the four types that Halogen supports: Level, Toggle, Selector, and Command. Once you create an external control in Halogen using the Control Systems dialog, you can then link it to any other linkable control of the same type in your Halogen configuration. This lets your external control system set and monitor any linkable parameter in your HAL configuration. Since the external controls can link to any other linkable control in a configuration, you have tremendous flexibility in the access that you provide to end users and seamless integration of your control system with internal controls, such as those in DSP block parameters and Rane Digital Remotes.

Each control that you configure for external access includes a unique number that allows external control systems to identify each control. For example, you might define a level control to be number *1* while a toggle control is number *2*.

## Connecting to a Halogen/HAL Control Server

The HAL system contains a Control Server to provide a way for your control system to connect and use the configured external controls. To integrate your external control system with a HAL system, your control system simply connects to the Control Server over TCP/IP. In the spirit of allowing you to configure and test your HAL system as much as possible without needing actual HAL hardware present, there are two ways that you can connect to a HAL Control Server to develop, test and use your external control system.

First, when developing and testing your system, the Halogen software includes the Control Server, which runs whenever Halogen is not connected to a HAL. Even though the configured HAL system hardware is not present, using the Control Server built into Halogen allows you develop your configuration, including all of the external controls, and test them with your external control system. Of course, because the actual hardware is not present, you cannot process audio or use physical devices such as Digital Remotes.

**NOTE:** To use the Control Server built into Halogen, connect to port 4996 on the TCP/IP address of the PC that is running Halogen.

The other way to use the Control Server is to connect to a HAL DSP Processor itself. This server is always available on a HAL at port 4996 on any of its TCP/IP addresses. Once you have created and applied a HAL configuration that contains external controls, these controls are available to your external control system via the Control Server. We recommend that you configure a static IP address on your HAL when using the HAL Control Server, so that the server is always available to your control system at a stable address.

**NOTE:** To use the Control Server built into HAL, connect to port 4996 on the TCP/IP address of the HAL.

## Communicating with a Halogen/HAL Control Server

Once connected to either the Control Server built into Halogen or to the server in an actual HAL, your external control system communicates with the Control Server using the HAL system External Control Message Protocol. This is an ASCII text based protocol that allows one or more control systems to access the external controls in your HAL configuration. Appendix A "HAL External Control Message Protocol" on page 40 fully defines these messages, but an example would be: <L&4&510>. This message is a 'set level' message that tells the HAL to set external level control number *4* to a new value of 51.0%. The protocol provides similar messages for getting and setting values for each of the four control types.

In addition to responding to messages it receives, the Halogen/HAL Control Server also sends messages to connected control systems whenever one of the external controls configured in HAL changes. This helps all of the connected external control systems remain up to date as values or links change.

## Using a Telnet Client to Test and Monitor the System

Because the message protocol is ASCII text, you can use a standard telnet client to connect to a Halogen or HAL Control Server and send/receive messages. This can be a great way to initially test your control system and to troubleshoot any problems that occur while developing and deploying your system. Appendix B "Using PuTTY to Test External Control Systems" on page 51 provides complete information for you to use a simple public domain Windows telnet client, PuTTY, with the Halogen/HAL Control Server.

# CHAPTER 3: Example HAL1x Configuration

The external control system example uses a HAL1x configuration that includes a set of external controls linked to processing block and preset controls that demonstrate some common uses of the end user external controls. There is at least one control of each type: Command, Level, Selector, and Toggle. We've also constructed this configuration using only inputs and outputs available on the HAL1x so that you can load and apply this configuration to any HAL1x without having to attach and configure other devices such as RADs or DRs.

*"Creating the Example Configuration" below* shows how to build the complete configuration from scratch. Of course, we also provide the configuration file for you so you don't have to go through all of the work, but it's your choice. You can either follow the directions below to create the configuration or open the file we provide and use the instructions below to learn about what's in the configuration.

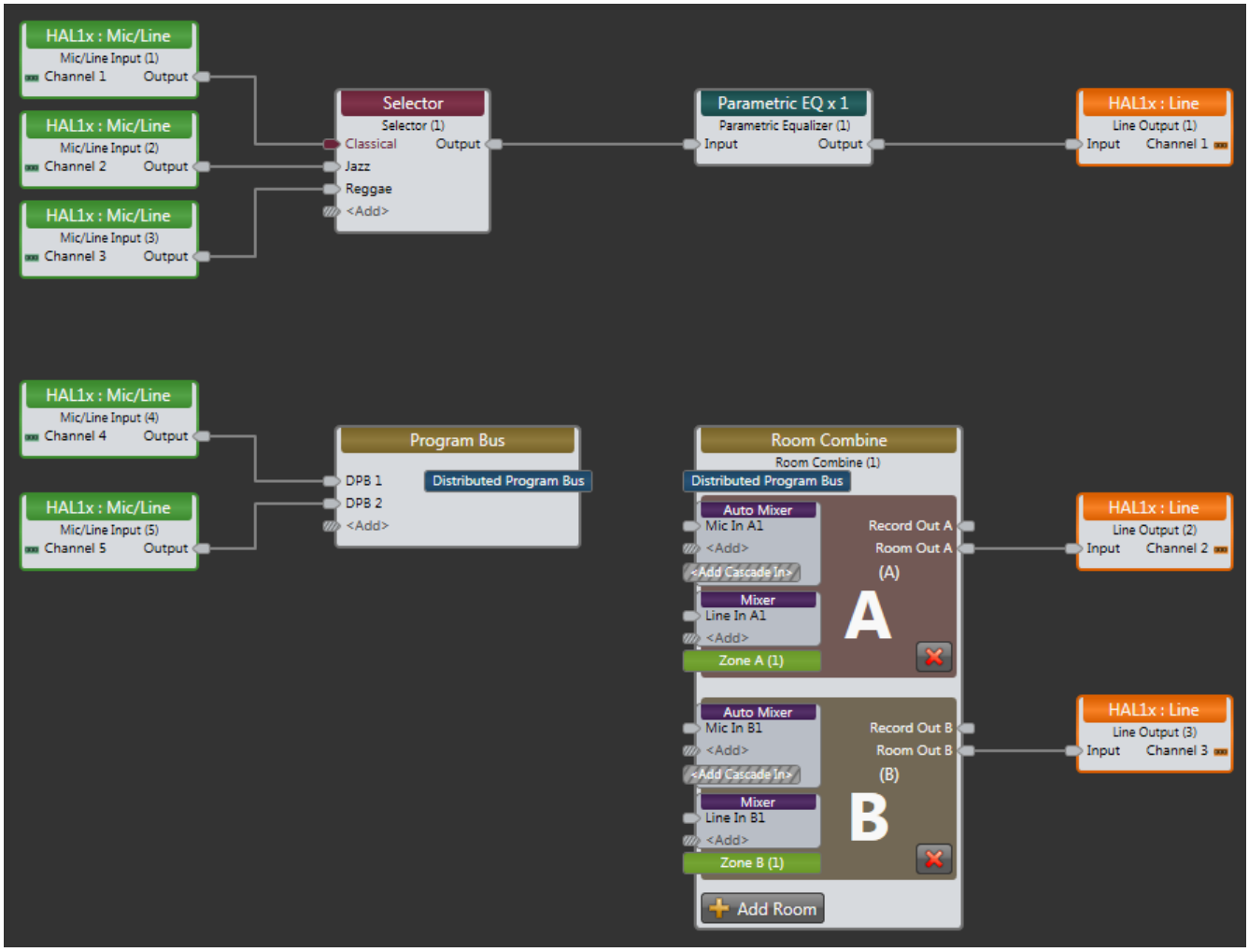
Alternatively, you can just read the "Configuration Summary" on page 15 to learn what external controls this configuration provides.

## Creating the Example Configuration

To get started, all of the work is done in the processing workspace, so click on the **Processing** tab in the upper right section of the Halogen application.

Click on the tab for the **I/O** palette and drag the first five HAL1x Mic/Line inputs from the palette and drop them on the left side of the processing map. In a similar way, drag the first three HAL1x Line Outputs from the palette and drop them on the right side of the processing map.

Next, fill in the middle processing area by clicking on the **DSP** palette tab and dragging four processing blocks to the processing map: Parametric EQ, Distributed Program Bus, Room Combine, and Selector. For the selector block, create one additional input and set the input names to Classical, Jazz, and Reggae. Also, delete room C from the Room Combine Processor by clicking the red 'X' in room C. Wire these together with your I/O blocks so that your processing map looks like this:



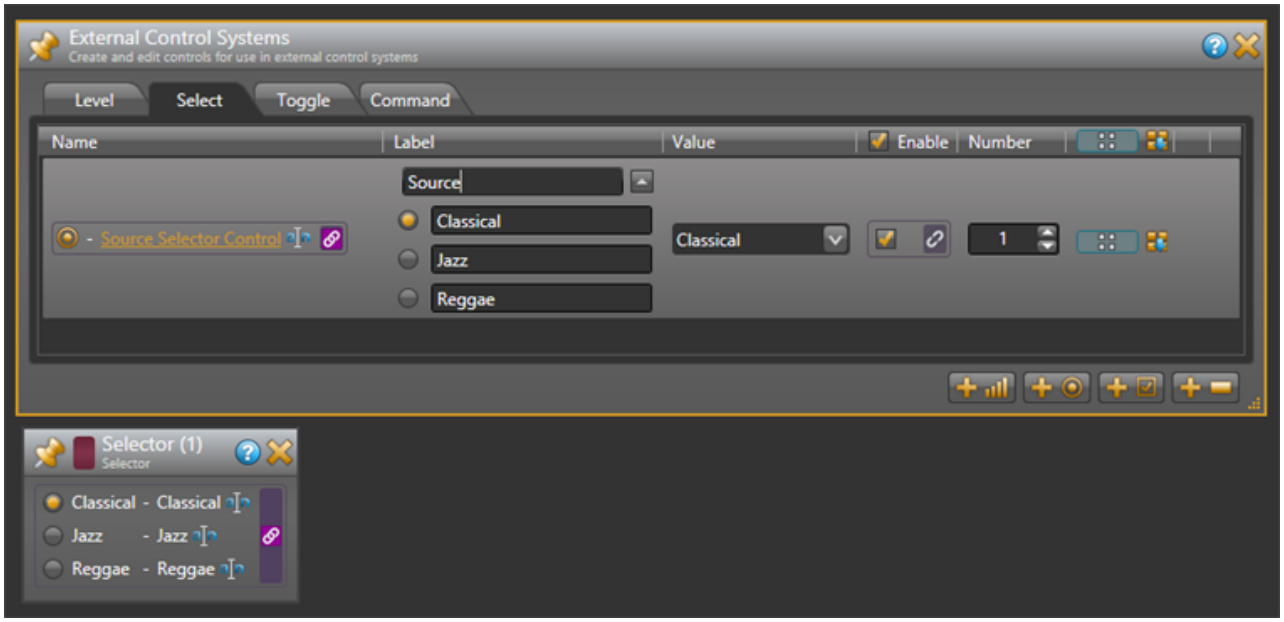

The top row of blocks provides audio for a lounge that has three program source inputs and some PEQ processing and the bottom row is a simple room combine configuration.

Next, we'll add some external controls using the External Control Systems dialog and link them to some of the controls available in the processing blocks we are using.

First, we'll create a control for the lounge input program selector:

1. Just above the upper left of the Processing Workspace, click on the **Control Systems** button then select **External Controls** to open the External Control Systems property dialog.
2. In the External Control Systems dialog, click on the **Select** tab.
3. Create a new Selector control by clicking on the button.
4. Change the name of this new control by clicking the edit icon and entering *Source Selector Control* followed by the Enter key.
5. Set its number to *1*.
6. Double click the Selector block in the Processing Map to open its property dialog

7. Link the new Selector control in the External Control dialog to the selector block's selector control. Do this by dragging the link icon in the External Control Systems dialog to the right of the control name and dropping it on the link icon for the Selector Block's selector control.
8. Change the Label in the Source Selector Control to "Source". Together, the selector property dialogs are as follows:

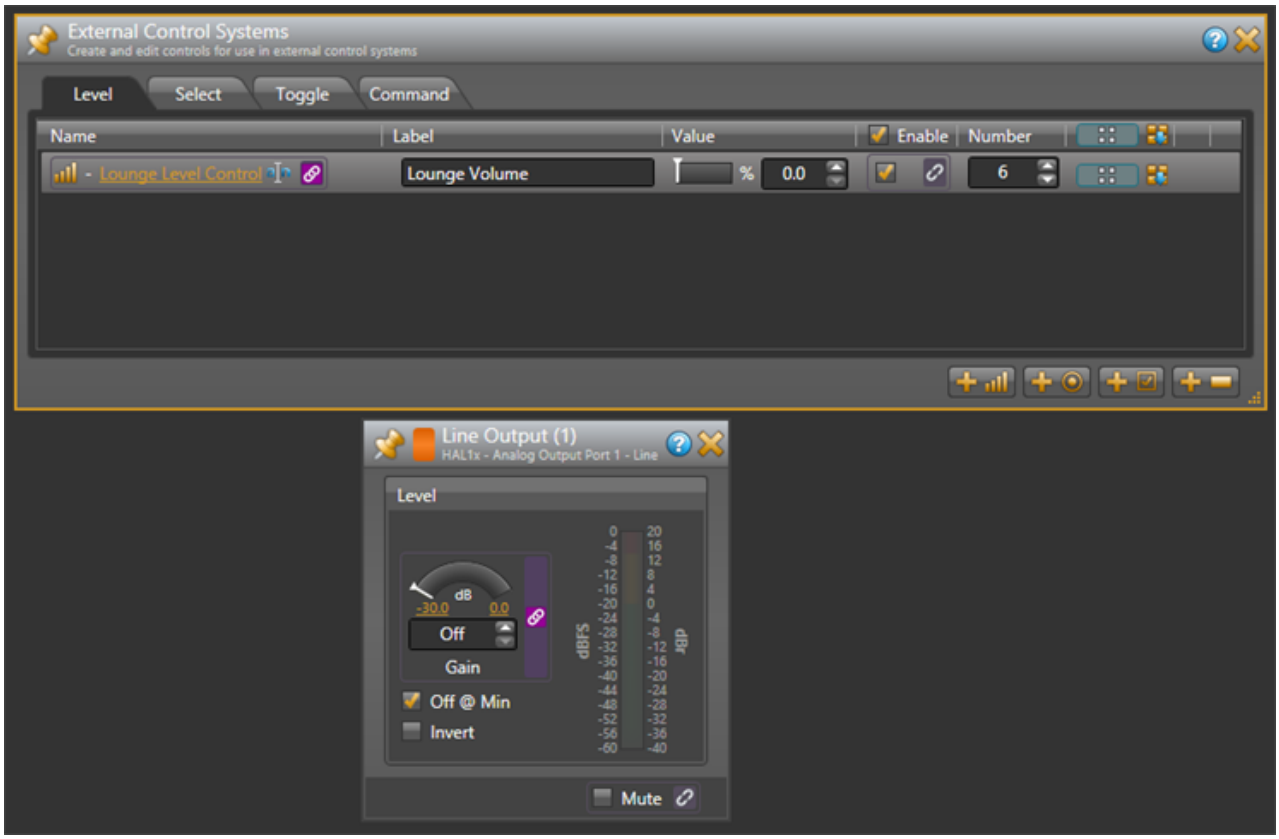



Your external control system can now get and set the lounge input source using the External Control Message Protocol.

Next, we'll add a control for the lounge volume level:

1. In the External Control Systems dialog, click on the **Level** tab.
2. Create a new Level control by clicking on the button.
3. Change the name of this new control to *Lounge Level Control* and set its number to *6*.
4. Open the property dialog for the lounge output block, HAL1x Line Output (1).
5. Link the new Level external control to the level control in the Line Output block property dialog. Do this by dragging the link icon in the External Control Systems dialog to the right of the control name and dropping it on the link icon for the Line Output Block's level control.

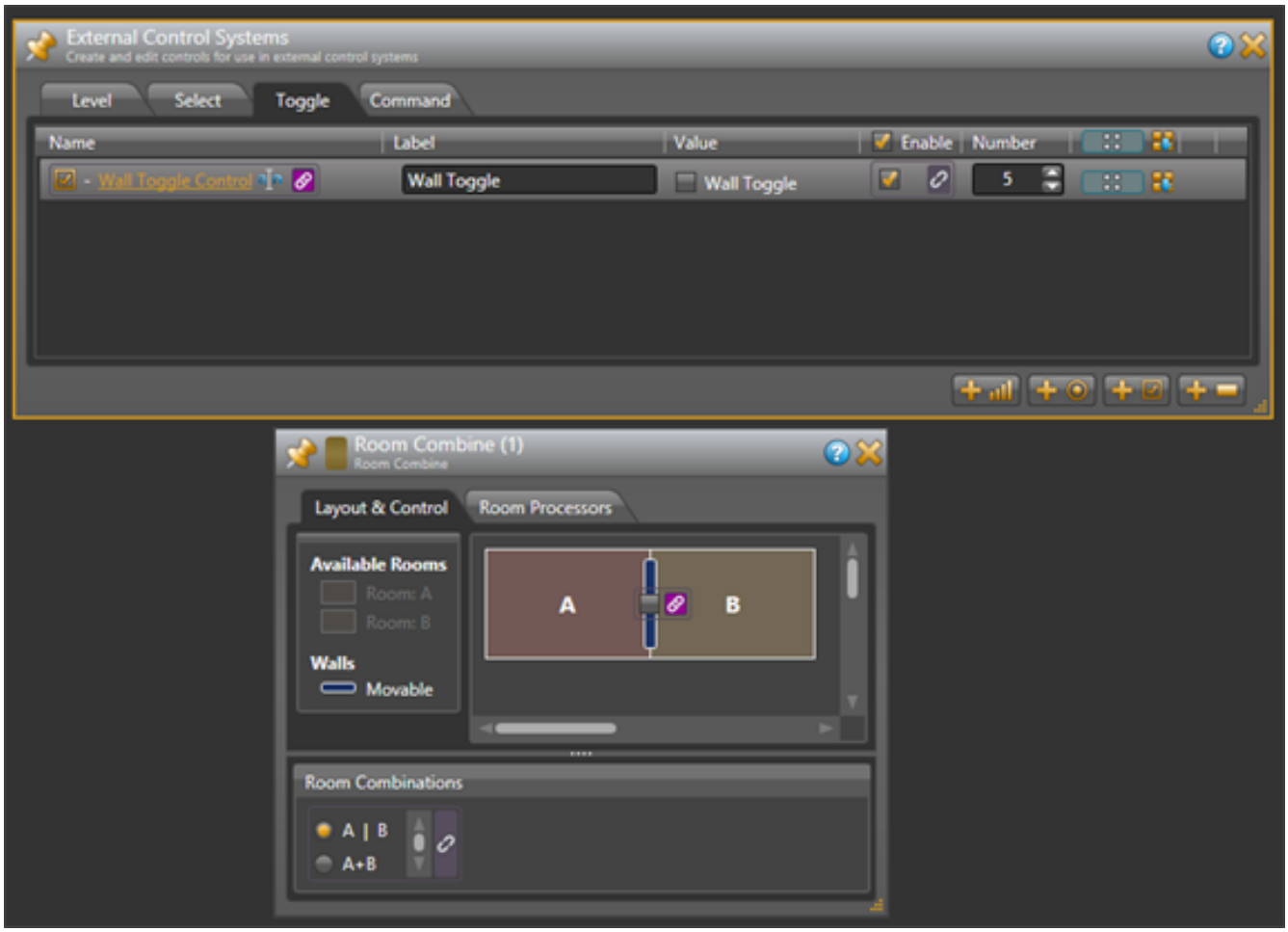
6. Change the Label for the Lounge Level Control "Lounge Volume". The dialogs appear below as follows:




In a Room Combine application, end users typically need to let the audio system know when they open or close a movable wall. To create a Toggle control for this:

1. Open the Room Combine Processor property dialog.
2. Drag both Room A and Room B onto the Layout & Control area and arrange them so that they are next to each other.
3. Add a movable wall between them by dragging the movable wall icon between the rooms until the wall highlights, then dropping the wall to create a wall toggle control.
4. In the External Control Systems dialog, click on the **Toggle** tab.
5. Add a new Toggle control by clicking on the button.
6. Change the name of the new control to *Wall Toggle Control* and set its number to *5*.
7. Link the new Toggle external control to the wall toggle control in the Room Layout & Control area. Do this by dragging the link icon in the External Control Systems dialog to the right of the control name and dropping it on the link icon for the wall toggle control between the two rooms.

8. Change the Label for the Wall Toggle Control to "Wall Toggle". The dialogs appear below as follows:



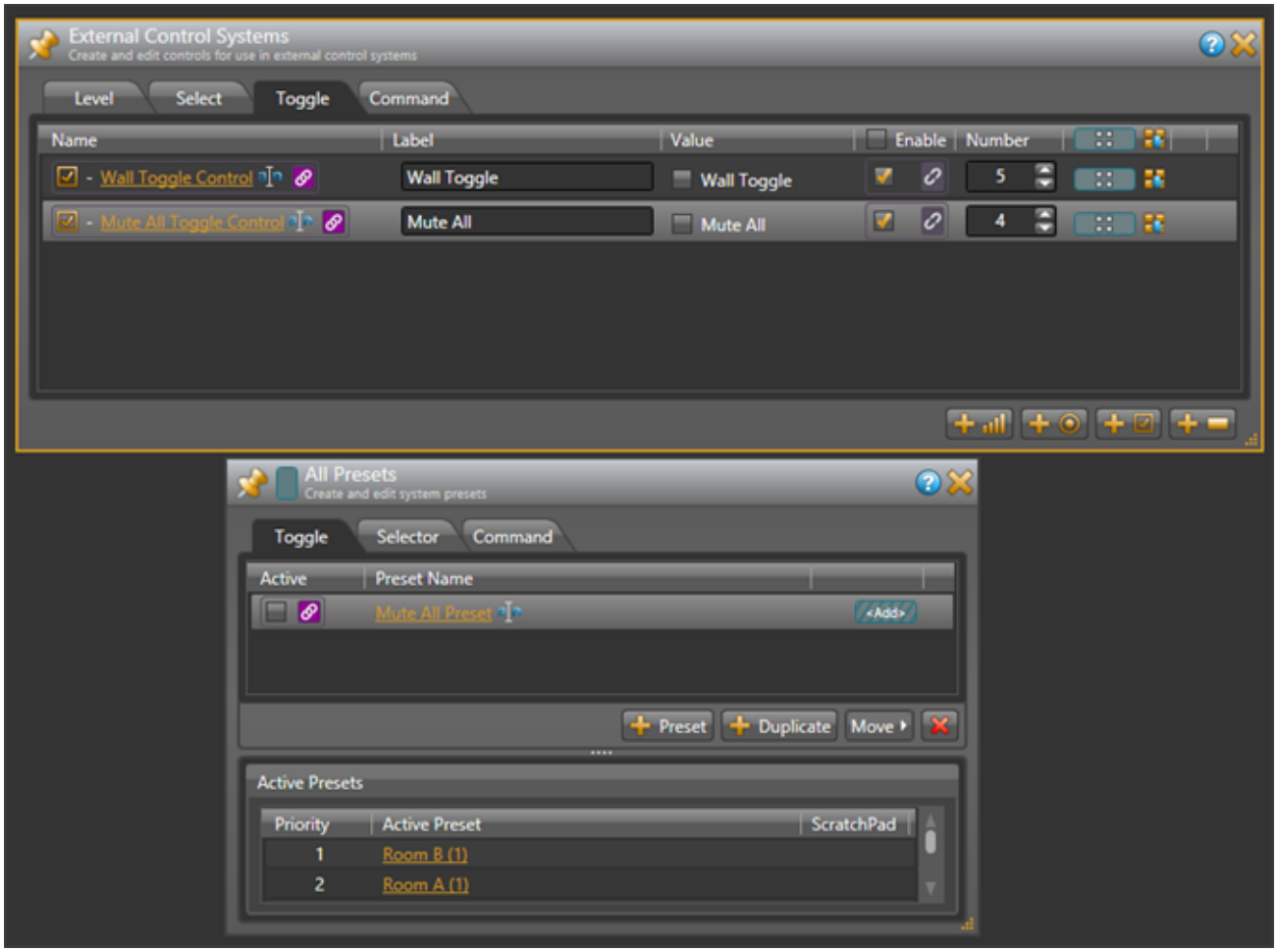

At this point, your external control system has access to three controls – a selector to get and set the lounge input program source, a level to control the lounge volume and a toggle to open and close the room combine wall.

Now we'll add some controls and link them to presets of various kinds to show how you can provide access to presets to your external control system.

First, we'll add a toggle control to mute all of the outputs in our configuration:

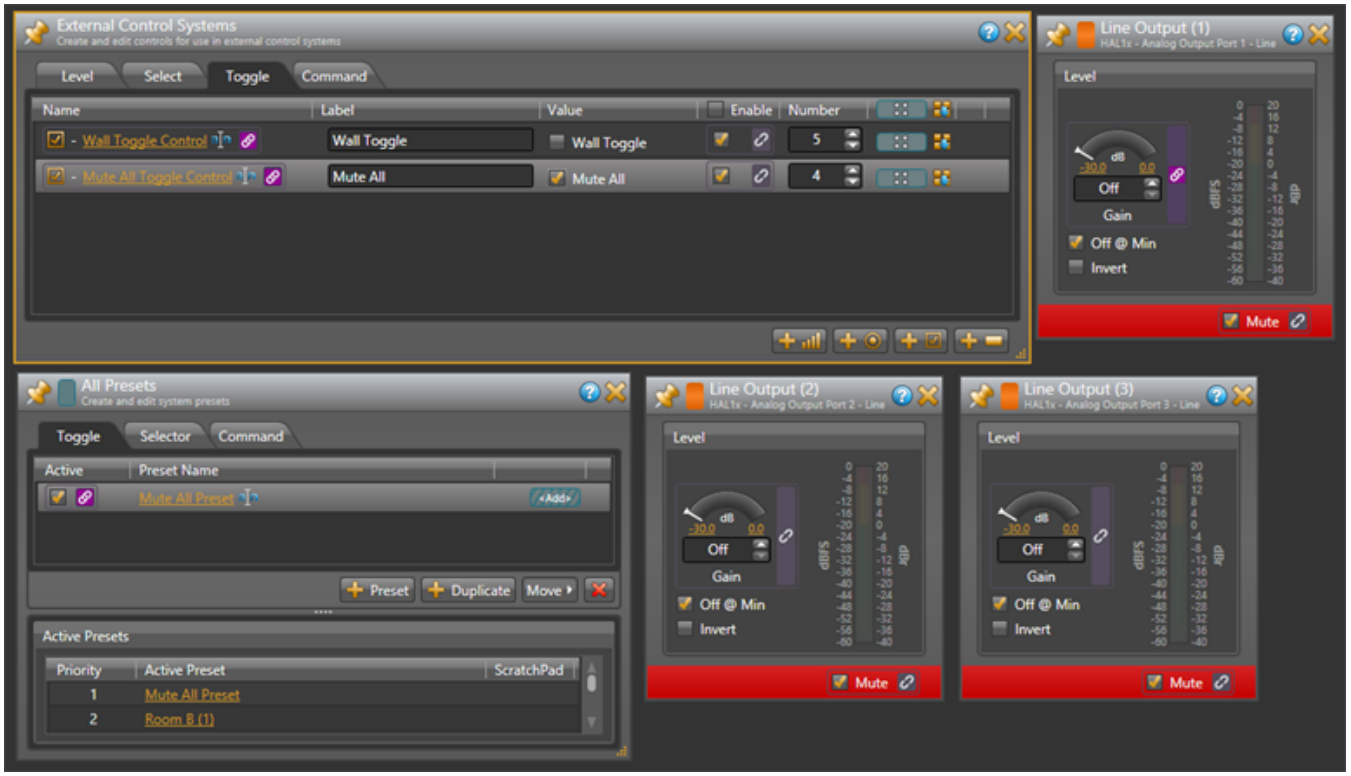
1. On the Processing Workspace Toolbar, click on the **Presets** button to open the All Presets property dialog.
2. In the All Presets dialog, click on the **Toggle** tab.
3. Add a new toggle preset by clicking on the +**Preset** button.
4. Rename this preset to *Mute All Preset*.
5. In the External Control Systems dialog, click on the **Toggle** tab if it is not already displaying the toggle controls.
6. Add a new Toggle external control by clicking on the button and set its number to *4*.
7. Rename this new control to *Mute All Toggle Control*.

8. Link the new Mute All Toggle Control to the toggle control for the Mute All Preset. Do this by dragging the link icon in the External Control Systems dialog to the right of the control name and dropping it on the link icon for the Mute All Preset's toggle control.
9. Change the Label for the Mute All Toggle Control to "Mute All". The dialogs appear below as follows:




10. Open the three HAL1x output blocks, set each of the mute checkboxes to checked (muted) and set their level controls all the way down (Off).
11. Add the three HAL1x output blocks to the Mute All Preset. To do this, hover your mouse over the HAL1x output block for the lounge. Drag the blue preset icon that appears and drop it on the blue rectangle for the Mute All Preset in the All Presets dialog. Repeat for the other two HAL1x output blocks.

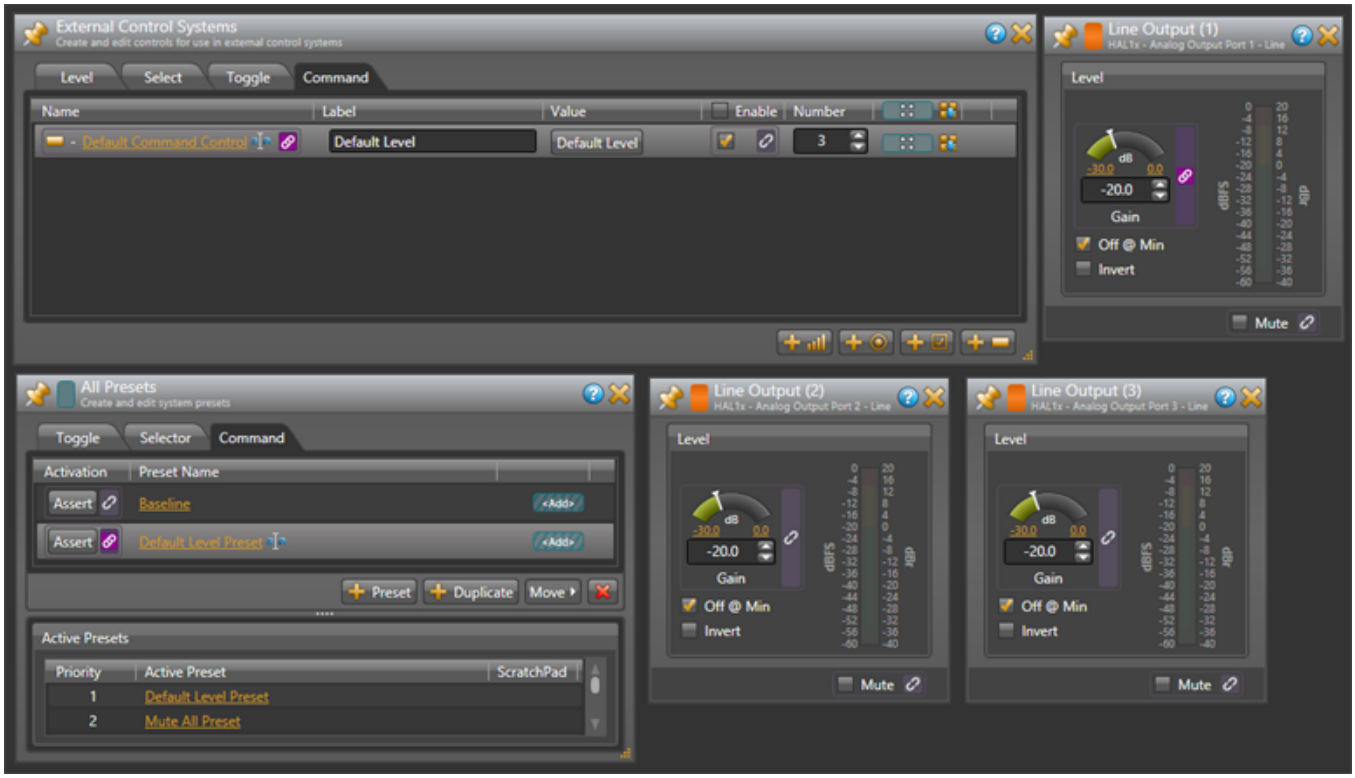
12. Test this by activating and deactivating the Mute All Preset from the All Presets dialog. All three HAL1x output blocks mute when the preset is active and un-mute when the preset is not active.




Now, we'll add a Command control to reset all outputs to a default operating state:

1. In the All Presets dialog, click on the **Command** tab.
2. Create a new command preset by clicking on the **+Preset** button.
3. Rename this new preset to *Default Level Preset*.
4. In the External Control Systems dialog, change to the **Toggle** tab if it is not already selected.
5. Make sure the **Mute All Toggle Control** toggle Value is un-checked.
6. Open the three HAL1x output blocks and set the level for each to *-20.0 dB* and un-check the mute toggle control.
7. Add the three HAL1x output blocks to the new Default Level Preset by dragging the blue preset icon next to each HAL1x output block and dropping them on the blue rectangle for the Default Level Preset in the All Presets dialog.
8. In the External Control Systems dialog, drag the blue preset icon for the Mute All Toggle Control and drop it on the blue rectangle for the Default Level Preset in the All Presets dialog. This ensures that the Mute All toggle is reset (unmuted) when the Default Level Preset is asserted.
9. In the External Control Systems dialog click on the **Command** tab.
10. Add a new Command external control by clicking on the button.
11. Rename this new control to *Default Command Control* and set its number to *3*.
12. Link the Default command control to the Default Level Preset's Assert button. Do this by dragging the link icon in the External Control Systems dialog to the right of the control name and dropping it on the link icon for the Default Level Preset's Assert button.

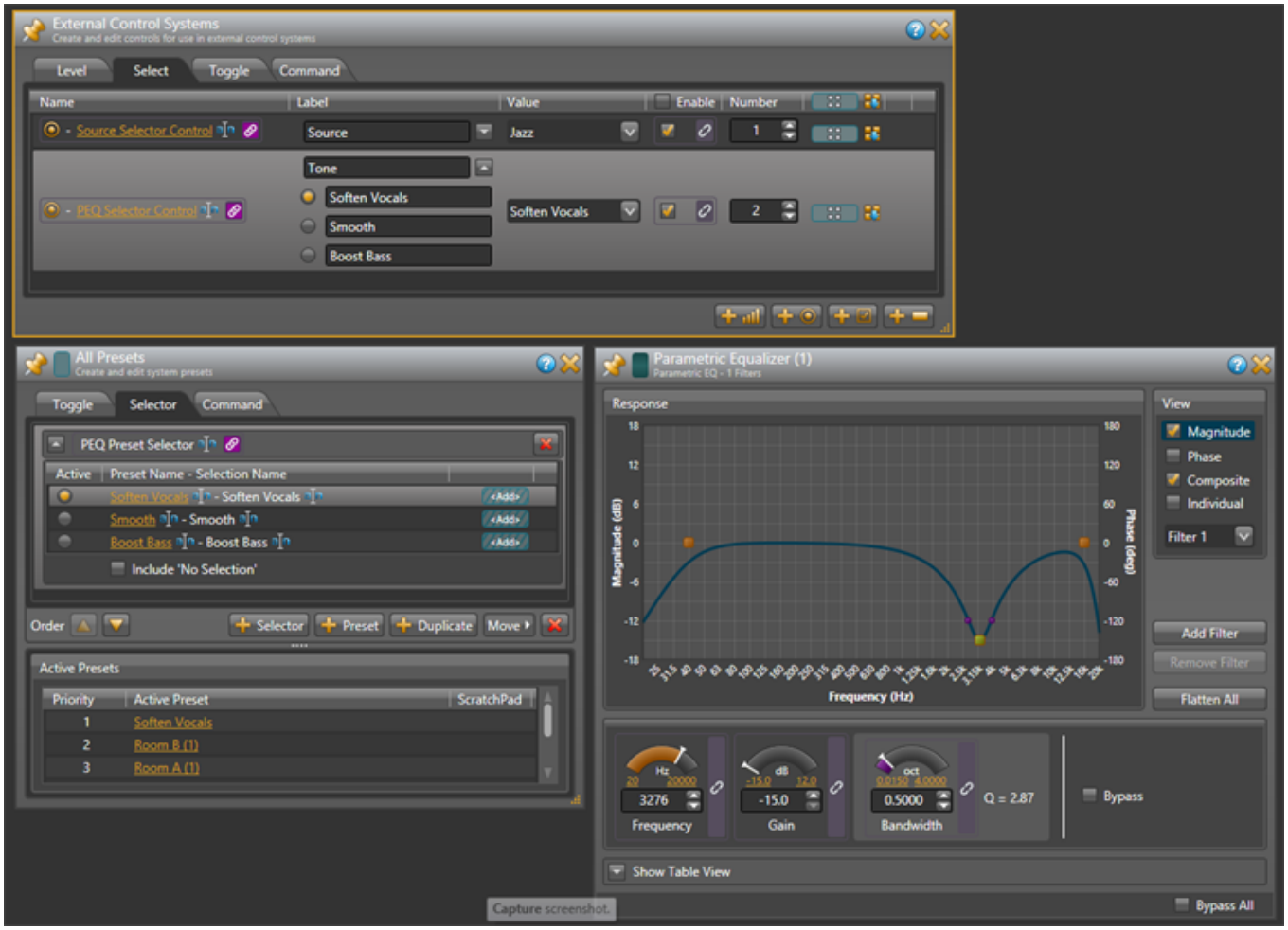
13. Change the Default Command Control's Label to "Default Level". The dialogs appear below as follows:




Finally, we'll create a selector preset to provide three options for the PEQ settings in the lounge

1. In the All Presets dialog, click on the **Selector** tab.
2. Create a new Selector preset by clicking on the **+Selector** button.
3. Rename this selector preset to *PEQ Preset Selector*.
4. Add three presets to this selector by clicking the **+Preset** button three times.
5. Change the names of these presets to *Soften Vocals*, *Smooth*, and *Boost Bass*.
6. Activate the *Soften Vocals* preset by clicking on the radio button to the left of its name in the All Presets dialog. Then uncheck the *Include 'No Selector'* checkbox at the bottom of the PEQ Selector Preset.
7. Open the property dialog for the PEQ block and set the Gain to *-15.0* and the Frequency to *3276*.
8. Add the PEQ block to the *Soften Vocals* preset – that is the first preset in the selector preset. To do this, drag the blue Preset icon next to the PEQ block in the processing map and drop it on the blue '<Add>' rectangle for the Soften Vocals preset.
9. Set the PEQ frequency to *1589* and add the PEQ block to the Smooth preset.
10. Set the PEQ frequency to *393*, the Gain to *+12.0* and add the PEQ block to the Boost Bass preset.
11. Test the preset selector by clicking on the Active radio buttons and observing the PEQ change.
12. In the External Control Systems dialog click on the **Select** tab.
13. Create a new Selector external control by clicking on the button.
14. Rename this new control to *PEQ Selector Control* and set its number to *2*.

15. Link the PEQ Selector Control to the PEQ Preset Selector control in the All Presets dialog. Do this by dragging the link icon in the External Control Systems dialog to the right of the control name and dropping it on the link icon for the PEQ Preset Selector control.
16. Change the PEQ Selector Control's Label to "Tone". The dialogs appear below as follows:




Wow, that's the whole configuration. Be sure to save your configuration before quitting Halogen.

## Configuration Summary

The example configuration provides the following external controls:

1. A Selector for lounge program source – control number 1

   There are three selections: Classical, Jazz, and Reggae

2. A Selector for lounge tone (PEQ) – control number 2

   There are three selections: Soften Vocals, Smooth, and Boost Bass.

3. A Level for lounge volume – control number 6

   This changes the output block's level for the lounge.

4. A Toggle to mute all audio outputs – control number 4

   This mutes all output blocks and sets their levels to Off. While setting the levels is not necessary, we've included it here to cause the lounge output level to change so the Halogen/HAL Control Server will send a 'set level' message.

5. A Toggle to open and close the wall of a room combine block – control number 5
6. A Command to reset audio outputs to default levels – control number 3

   This sets the three outputs to -20.0dB and un-mutes them. In addition, it sets the Mute All toggle control to unchecked so the state is consistent with the outputs being un-muted.

# CHAPTER 4: Integrating Stardraw Control Systems with HAL

## Overview

This guide describes an example Stardraw Control application and driver that communicates with a Halogen/HAL Control Server. It is designed to be used in conjunction with the files found in the Stardraw Control Support Package. The contents of this support package include:

- StardrawControlSystems_Guide.pdf - The guide you are reading now
- ControlSystemSample.hal - Halogen configuration file intended for loading in Halogen or your HAL1x
- HAL1 Stardraw Example Project.s03 - Stardraw Control Project
- HAL1 Example Driver.cs - Stardraw HAL1 driver source code

The Stardraw Control Support Package is installed with the Halogen software and can be accessed from the Windows Start Menu under Halogen -> Guides. If you want to access the files directly they are available on the product DVD or from the Halogen install directory:

- C:\Program Files (x86)\Rane Corporation\Halogen\Guides\Support Packages\Stardraw Control directory

You can also download the most up-to-date version of this support package from the Rane website http://rane.com/hal).

**NOTE:** You do not need to connect your Stardraw Control application to an actual HAL in order to test your control system. A PC running Halogen can simulate all the feedback your application would get from a HAL running the same configuration.

The HAL system has four types of controls: toggles, commands, levels and selectors. External TCP/IP control systems, including a Stardraw Control application, can access all four types in any combination depending on the application. The previous section described how to create the sample configuration for use with the example Stardraw driver and application. This section walks you through the structure of the Stardraw HAL1 TCP/IP driver, which communicates with a Halogen/HAL Control Server and also provides a representation of the controls to the Stardraw Control application. We're using 'Halogen/HAL Control Server' in this section to remind you that you can run your Stardraw control system in one of two ways:

- With the Control Server in Halogen when it is not connected to a HAL1x. This lets you develop and test the system when you don't have a HAL1x available.
- With the Control Server in an actual HAL1x DSP Processor connected to your network.

In addition, this section describes an example Stardraw Control application that uses the driver, focusing on each of the control types one-by-one. We use built-in Stardraw Control panels and controls to provide controls for the example HAL configuration. This application includes:

- A multi-position control to select the lounge input program source
- A multi-position control to select the tone setting for the lounge audio
- A command button to reset all audio outputs to default levels
- A toggle control to mute all audio outputs of the audio system

- A toggle control to open and close a wall in the room combine block
- A slider control to adjust the volume of the lounge level

This section concludes with a set of troubleshooting topics that cover the most common problems that could occur when using the example Stardraw application with Halogen or HAL1x.

# Checklist

To use the files from the Stardraw Control Support Package and follow along with the examples you will need:

- A Windows desktop or laptop with an Ethernet port and the latest Rane Halogen (1.1.0 or higher) and Stardraw Control applications installed.

Although you will ultimately want to connect your Stardraw control system to a HAL you will see that most of the integration and testing can be accomplished without a HAL. To apply the sample configuration to a HAL1x and connect the Stardraw example application to the HAL1x, you will also need:

- A Rane HAL1x DSP Processor
- An Ethernet switch and CAT 5 cables for the PC and HAL1x.

# Getting Started

## Overview of Stardraw Control Application Development

The overall approach for developing a Stardraw Control application is to:

- Select drivers for devices you want to use and drag them to the topology canvas
- Wire the drivers together according to application's needs and available communications ports
- Create a custom user interface by dragging panels and controls to a form
- Connect the controls to the driver by creating actions for available driver and control events

You do all of this in the Stardraw Control authoring application, which also allows you to run your control application under a debugger, test your application, and make changes as necessary.

Once your Stardraw control application is ready for end users, you compile it to an executable file and deploy it to the target system.

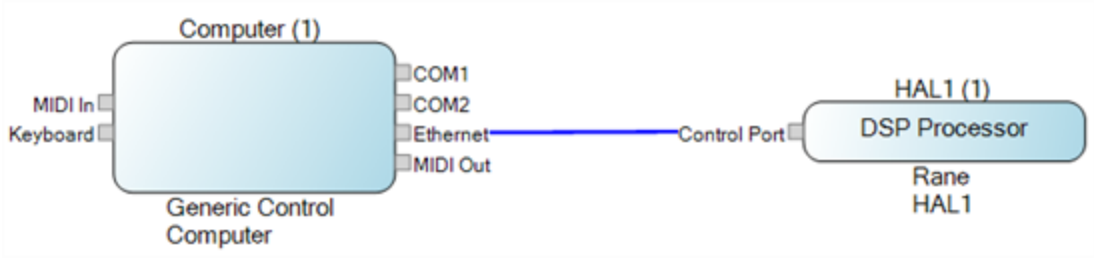
## Importing the Example Stardraw Project

To get the most out of this guide, you should import the example into the Stardraw Control 2010 authoring application. The example project includes the example application built on the example HAL1 driver.

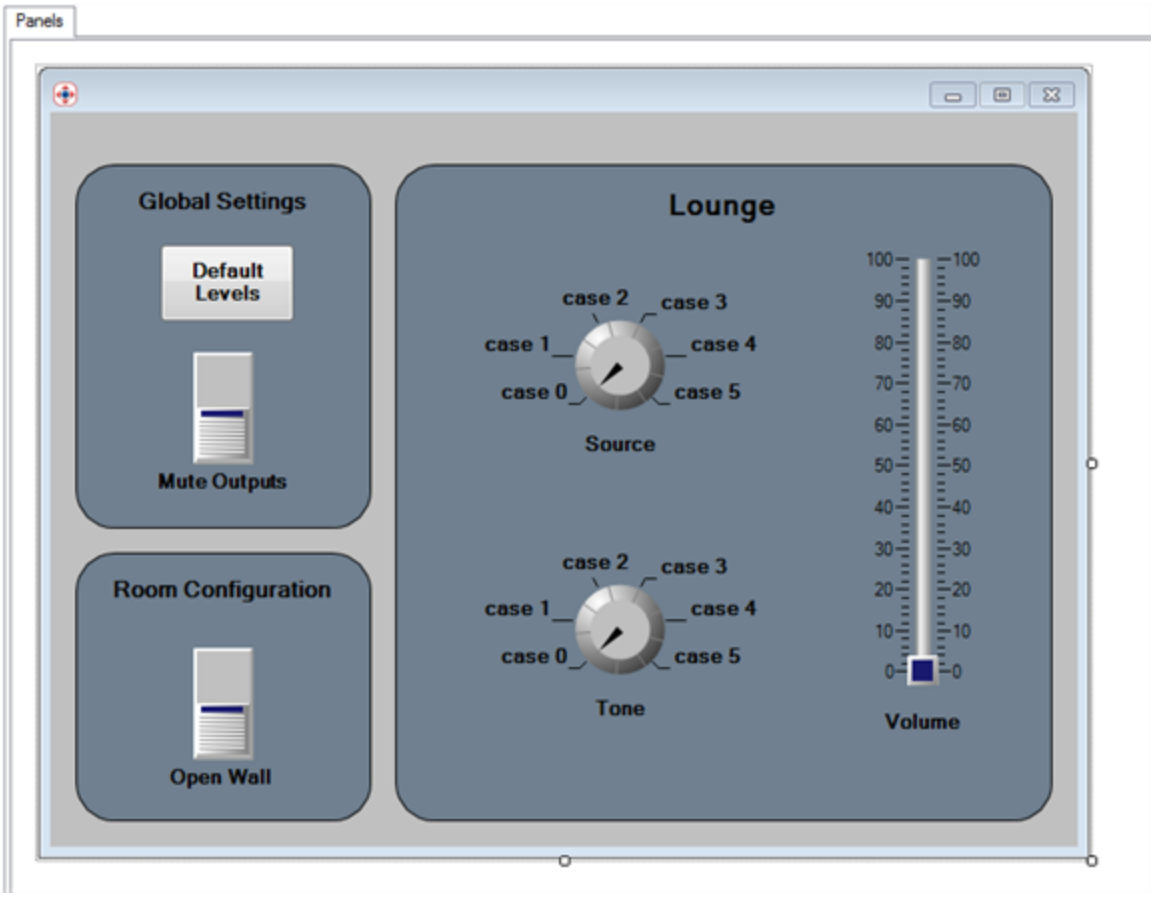
To import the example project, do the following:

1. In the Stardraw Control 2010 authoring application, select the File menu and click on Import Project.
2. In the dialog that appears, navigate to the Halogen Stardraw Example directory and click on the HalExampleProject.s03 file, then click the Open button.
3. When the Project Details window appears, enter a suitable Project Name and Project Description.
4. Click on the Panels tab and check that the Executable Path is valid. If not, change it to a path that exists on your PC.
5. Click OK.
6. Save your new project by clicking on the floppy disk icon in the Stardraw toolbar.

When you are finished, the Stardraw Control topology will look similar to this:




Switch to the Forms tab in the Stardraw authoring application by clicking on the **Forms** tab at the bottom left of the Stardraw window and you will see that the basic application form looks like this:




You can run the application and see how it works by doing the following:

1. Start Halogen by double clicking on the Halogen icon of your PC.
2. After Halogen starts, open the example HAL1x configuration file, ControlSystemSample.hal, that resides in the Halogen Stardraw Example directory. Do this by clicking on the Open button in Halogen and navigating to the directory and selecting the file.
3. Run the Stardraw application under the debugger by clicking on the blue arrow in the Stardraw toolbar (Run Program.)

4. When the Stardraw panel appears, use the controls to change the setting of the external controls that are included in the HAL configuration.

For example, open the HAL1x lounge output block in the Halogen processing map. Also open the Control Systems property dialog, change to the Level tab and click on the Lounge Level Control hyperlink. If you watch the three level controls (Stardraw Volume, Lounge Level Control, and Line Output (1)), you will see they all track each other, regardless of which one you actually change on the screen.

If the Stardraw application cannot connect to Halogen or the controls do not work correctly, see the Troubleshooting topics at the end of this section.

# Understanding the Example HAL Driver

The example Stardraw HAL driver serves as the bridge between the Stardraw application and the Halogen/HAL Control Server. Its purpose is to provide suitable properties and events for the Stardraw application and to handle Halogen/HAL Control System messages – both sending them to the Halogen/HAL Control Server when the Stardraw application changes a control and receiving messages from Halogen/HAL and making sure the Stardraw application knows about them.

This section describes the example Stardraw HAL driver in two parts. The more generic part is first, and discusses how the driver communicates with the Halogen/HAL Control Server. This part is suitable for use as the basis of any HAL driver. The second part describes methods, properties and events that the driver provides for the Stardraw application and how these are hooked to the HAL communications driver portions. These methods, properties and events are specific to the external controls configured in the HAL1x example configuration file and allow the Stardraw application to use them.

## Part 1 - Communicating with the Halogen/HAL Control Server

### Sending Messages to Halogen/HAL

Sending a message to the Halogen/HAL Control Server is very easy. The HAL1 driver is built upon a class, `TcpInPortInstance`, which Stardraw provides for basic communications with TCP/IP based devices. The `TcpInPortInstance` class provides a method, `WriteLine(string)`, that sends the string argument to the connected TCP/IP device, in this case Halogen or a HAL1. The example HAL1 driver includes a set of utility methods that each create a HAL external control message and use `WriteLine` to send it to Halogen/HAL. There is a method for each type of message that the driver can send to Halogen/HAL (except for the `<ping>` message, which we don't use in the example driver). These are:

| Method | External Control Message |
|---|---|
| **Level** | |
| `setLevel(uint controlID, int value)` | 'set level' |
| `requestLevel(uint controlID)` | 'get level' |
| **Toggle** | |
| `setToggle(uint controlID, int value)` | 'set toggle' |
| `requestToggle(uint controlID)` | 'get toggle' |
| **Command** | |
| `executeCommand(uint controlID)` | 'set command' |
| **Select** | |
| `setSelection(uint controlID, int value)` | 'set selection' |
| `requestSelection(uint controlID)` | 'get selection' |

| Method | External Control Message |
|---|---|
| `requestSelectionLink(uint controlID)` | 'get selection link' |
| `requestSelectionName(uint controlID, int selectionNumber)` | 'get selection name' |

See Appendix A for the complete definition of the HAL external control messages and their formats.

To see how these methods work, look at the first one in the list, `setLevel()`:

```
/// <summary>
/// Send a set level message
/// </summary>
/// <param name="controlNumber">The HAL control system number for the
control</param>
/// <param name="value">The level value to set</param>
private void setLevel(uint controlNumber, int value)
{
  string cmd = string.Format("<L&{0}&{1}>",
                             controlNumber, value);
  WriteLine(cmd);

  Logger.Debug("Level - set command sent: " + cmd);
}
```

This method first creates a message string, cmd, formatted according to the HAL external message protocol using the supplied parameters for the control number and value. For example, if the driver called the method like this:

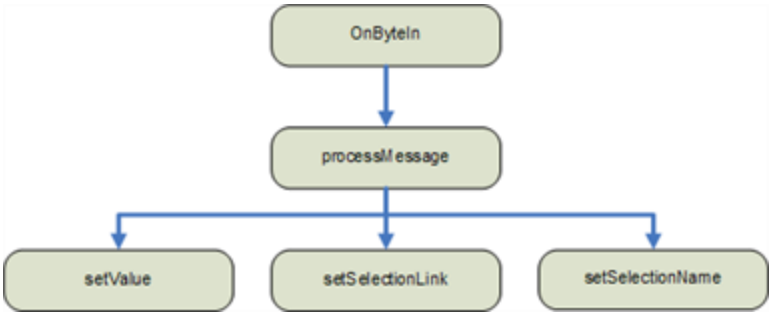
```
setLevel(6, 800);
```

the method would send the HAL external control message `<L&6&800>` to the Halogen/HAL Control Server, telling it to set level control number *6* to a value of *800* (80.0%). You will also notice that `setLevel()` also logs a debug string to the Stardraw Logger, which appears in the Debug window when running the Stardraw application under the debugger.

### Receiving and Processing Messages from Halogen/HAL

Receiving and processing messages that Halogen or HAL sends to the Stardraw driver is somewhat more difficult. This is because the driver not only receives messages, but it must also decode them and let the Stardraw application know about the information contained within them.

The methods that the example driver uses for this are:




The driver overrides Stardraw's `TcpInPortInstance.OnByteIn()` method, which runs whenever the connected Halogen/HAL Control Server sends a message byte to the Stardraw application. `OnByteIn()` handles each character as it is received and when it has recorded a complete message, it passes it off to the `processMessage()` method.

The `processMessage()` method decodes the message and calls one of three other methods depending upon the type of message it decodes:

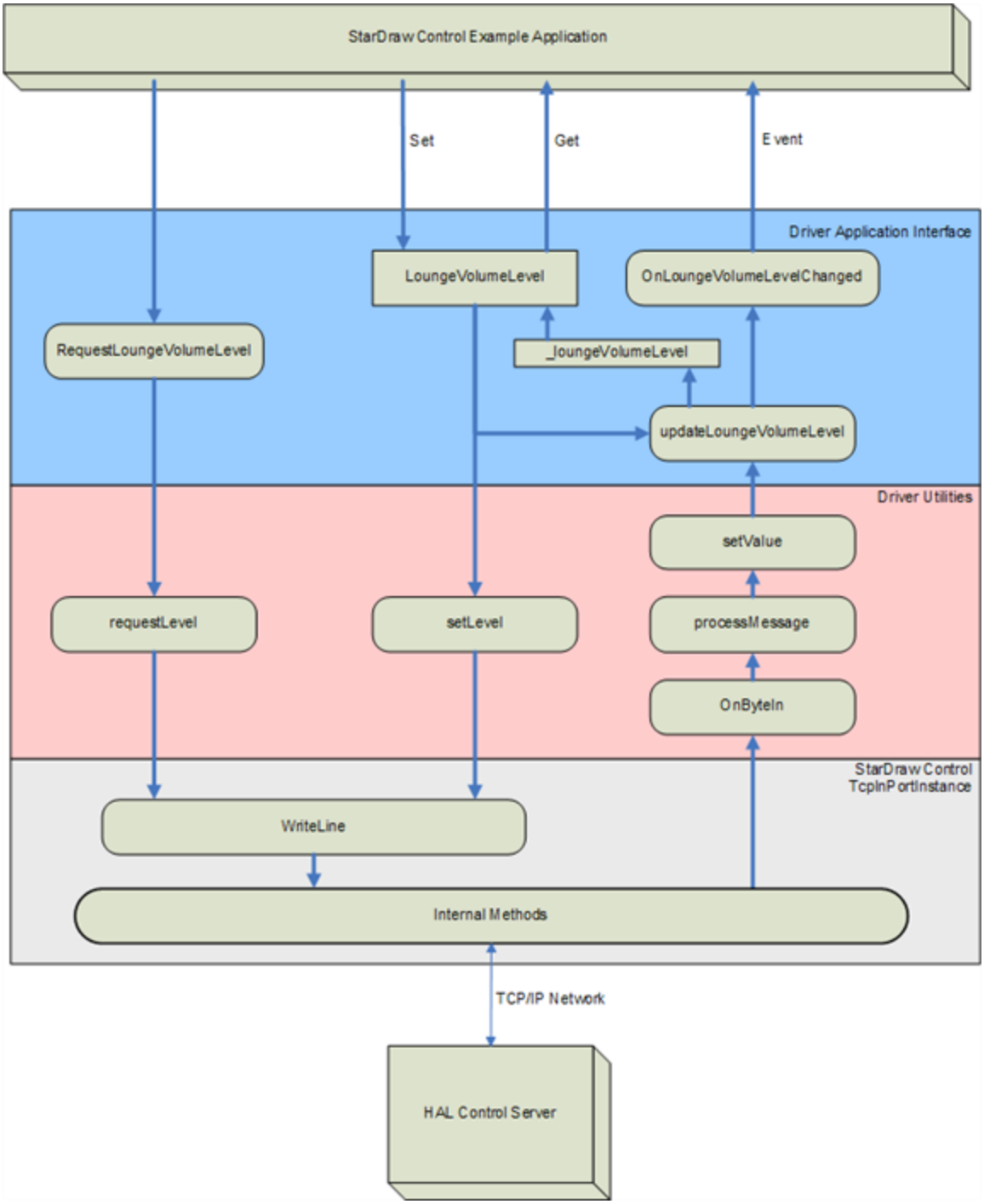
- `setValue()` for 'set toggle', 'set selector', and 'set level' messages;
- `setSelectionLink()` for 'set selection link' messages
- `setSelectionName()` for 'set selection name' messages.

These three 'set' messages update the state of the internal driver properties and variables according to their areas of responsibilities. The `setValue()` method updates all of the Level, Toggle, and Selector values while `setSelectionLink()` updates the number of selections in each selector and `setSelectionName()` updates of the name for a selection item in a selector. In the next section you will see how this relates to the properties and events that the driver exposes to the Stardraw application.

## Part 2 - Connecting to the Stardraw Application

To make it as straight-forward as possible for the Stardraw Control example application, the driver provides a set of properties that allow the application to get and set the value of each external control included in the example Halogen configuration. In addition, the driver also provides several "On…Changed" events which notify the Stardraw application whenever it receives a message that requires an update of the state of a control.

For example, the driver structure for the lounge volume level control and how it interacts with both the Stardraw application and the Halogen/HAL Control Server is as follows:




This builds upon the information in the previous section, which describes everything in the 'Driver Utilities' layer and below. The Driver Application Interface layer provides a property, `LoungeVolumeLevel`, that allows the Stardraw Control application to get or set the value of the Lounge Volume control. When the application 'gets' the `LoungeVolumeLevel`, the driver simply returns its current value, `_loungeVolumeLevel`, stored locally inside the driver. When the application 'sets' the `LoungeVolumeLevel`, for instance when the end user adjusts the slider control, the driver sends the new value to the HAL Control Server by calling the `setLevel` utility

method with the proper control number and desired value. As described before, `setLevel` creates a new message for the change and sends it to Halogen/HAL by calling the built-in `WriteLine` method.

As part of setting a new lounge volume level, `LoungeVolumeLevel` also calls the `updateLoungeVolumeLevel` method, which updates the locally stored value of the lounge volume level, `_loungeVolumeLevel`, and also sends an `OnLoungeVolumeLevelChanged` event for to the Stardraw application. This lets anything registered to receive this event know that it should update to the new value of `LoungeVolumeLevel`.

When the Stardraw application wants HAL to send the current value of the lounge volume control, it calls `RequestLoungeVolumeLevel`, which in turn calls `requestLevel`, causing the driver to send a 'get level' message to the Halogen/HAL Control Server asking for the value. When Halogen/HAL responds, it sends a 'set level' message to the Stardraw driver, which the driver handles starting with `OnByteIn`, eventually resulting in an `OnLoungeVolumeLevelChanged` event, letting the application know that a new value for the lounge volume level is available.

## The Complete Driver Application Interface

The following table shows all of the driver methods, properties, and events available to the Stardraw control application for the example HAL configuration. These all work similarly to the lounge volume level control that we described above.

| Methods | Properties | Events |
|---|---|---|
| **Lounge Volume Level** | | |
| RequestLoungeVolumeLevel | LoungeVolumeLevel | OnLoungeVolumeLevelChanged |
| **Mute All Toggle** | | |
| RequestMuteAllToggle | MuteAllToggle | OnMuteAllToggleChanged |
| **Wall Open Toggle** | | |
| RequestWallOpenToggle | WallOpenToggle | OnWallOpenToggleChanged |
| **Lounge Source Selector** | | |
| RequestSourceSelection | SourceSelection | OnSourceSelectionChanged |
| RequestSourceSelectionMaximum | SourceSelectionMaximum | OnSourceSelectionMaximumChanged |
| RequestSourceSelectionNames | SourceSelectionNames | OnSourceSelectionNamesChanged |
| **Lounge Tone Selector** | | |
| RequestToneSelection | ToneSelection | OnToneSelectionChanged |
| RequestToneSelectionMaximum | ToneSelectionMaximum | OnToneSelectionMaximumChanged |

| Methods | Properties | Events |
|---|---|---|
| RequestToneSelectionNames | ToneSelectionNames | OnToneSelectionSelectionNamesChanged |
| **Default Levels Command** | | |
| ExecDefaultLevelsCommand | | |
| **Form Initialization** | | |
| RequestAllControlValues | | |

# Exploring the Stardraw Example Application

## Overview

The example application includes a variety of controls that correspond to the set of external controls included in the HAL1 example configuration. This section describes each of the controls contained in the Stardraw form and how they connect to and use the example driver's methods, properties, and events.

## Toggles

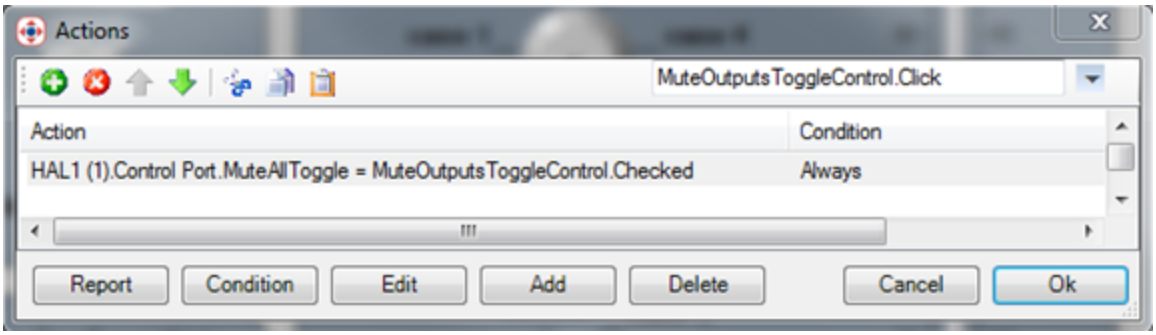
The example application provides two toggle controls, one for muting all audio outputs and one to open and close the Room Combine block's wall toggle. In the Stardraw form, both of these use the same built-in **Switch** control from the Stardraw **Toolbox Controls** palette.

### Mute Outputs Toggles

Hooking up the **Mute Outputs** switch to the driver is a matter of mapping two events to actions which provide bi-directional communication between the Stardraw application and the HAL Control Server.

First, when the end user changes the switch setting, the Stardraw application generates a **Click** event for the **Mute Outputs** switch, which is named *MuteOutputsToggleControl*. In the application, this event is mapped to the action which sets the HAL1 driver's *MuteAllToggle* property to the current state of the Stardraw *MuteOutputsToggleControl*. This means that whenever the end user changes the switch setting, the Stardraw application sends a corresponding change to the Halogen/HAL Control Server.

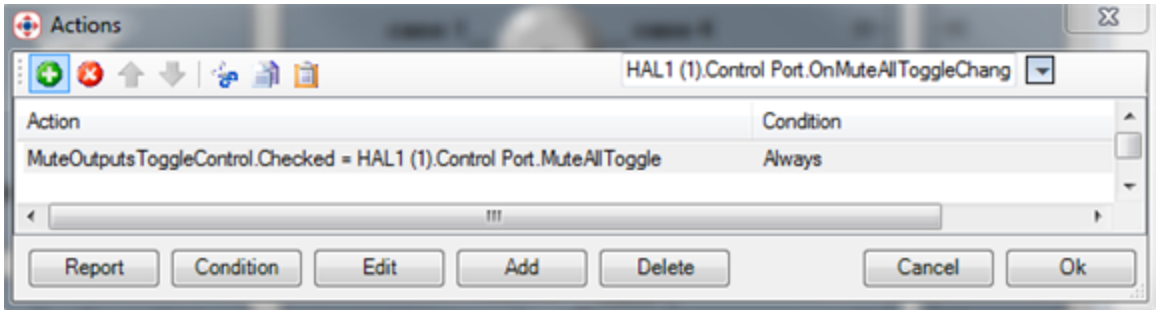
To see this mapping, double-click on the **Mute Outputs** switch in the Stardraw form. This brings up the **Actions** dialog for that control (named *MuteOutputsToggleControl*):



In this dialog, the event is *MuteOutputsToggleControl*.**Click**, shown in the upper right. For this event, the dialog shows the set of actions that run when the event occurs. In this case the application sets the HAL1 *MuteAllToggle* driver property to the current **Checked** value of the *MuteOutputsToggleControl*. As we saw in the driver description, this causes the driver to send a 'set toggle' message to the HAL Control Server for the **Mute All Toggle Control** configured in the Halogen Control Systems dialog.

The other event that needs to be hooked up for the switch control is the HAL1 *OnMuteAllToggleChanged* event. This occurs when the HAL1 driver receives a 'set toggle' message for the *Mute All Toggle Control*. To keep the Stardraw application 'in-sync' with the HAL, this event maps to an action that updates the **Checked** value of the *MuteOutputsToggleControl*.

You can see this in the Actions dialog for the *MuteOutputsToggleControl* by changing the event in the dialog drop-down to HAL1 *OnMuteAllToggleChanged*. Do this by clicking on the drop-down arrow and then traversing the popup menus: Devices -> HAL1 (1) -> Control Port and then selecting *OnMuteAllToggleChanged*. The dialog will change to:




This shows the action that runs when the HAL1 driver raises the *OnMuteAllToggleChanged* event. In this case the action updates the Stardraw *MuteOutputToggleControl*'s **Checked** state to the current value of the HAL driver's *MuteAllToggle* property.

### Open Wall Toggle

The configuration of the **Open Wall** switch is very similar to the **Mute Outputs** switch configuration. In this instance, however, the name of the form control is *OpenWallToggleControl*, and the application maps this control's **Click** event to an action that sets the HAL1 driver's *WallOpenToggle* property to the current **Checked** value of *OpenWallToggleControl*.

In addition, the application maps the HAL1 *OnWallOpenToggleChanged* event to an action that updates the form's *OpenWallToggleControl* to the current value of the HAL1 driver's *WallOpenToggle* property.

As before, you can see these events and associated actions by double clicking on the form's **Open Wall** switch.

## Commands

To unmute all audio outputs and reset them to default levels, the example HAL1 configuration provides a Command control. The Stardraw application implements this function with the **Default Levels** button, which is a standard button from the Stardraw **Toolbox Windows** palette.

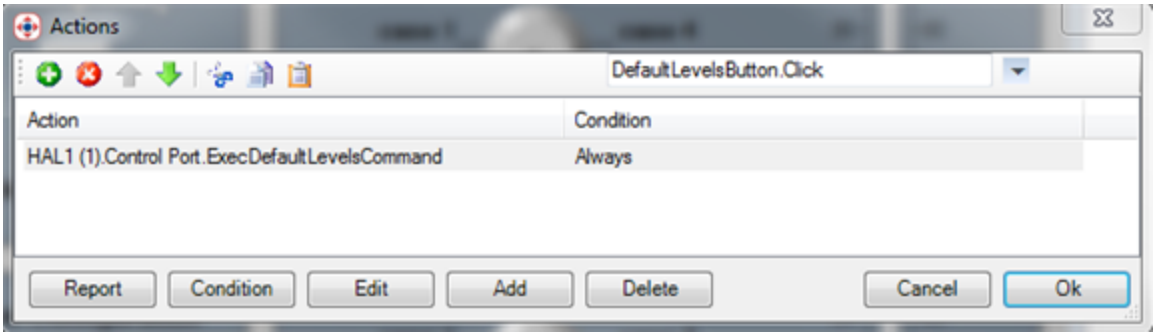
### Default Levels Button

The **Default Levels** button is unidirectional – it sends messages to the Halogen/HAL Control Server but does not receive them. This is because commands occur in a moment of time – when the user clicks on the command but-

ton. There is no state to maintain in the form and so receiving and handling command messages that the Halogen/HAL Control Server sends is optional.

When the user clicks on the **Default Levels** button, the Stardraw application generates a **Click** event for the form's *DefaultLevelsButton* control. The application has mapped this event to an action which calls the HAL driver's *ExecDefaultLevelsCommand* method. This in turn causes the driver to send a 'command' message to the Halogen/HAL Control Server which fires the *Default Command* Control configured in the HAL example configuration.

To see the event and associated action, double click on the **Default Levels** button. This brings up the **Actions** dialog for the *DefaultLevelsButton*:




In this dialog the event is *DefaultLevelsButton*.**Click**. When this event occurs, the action calls the *ExecDefaultLevelsCommand* method in the HAL1 driver.

## Levels

The end user of the Stardraw example application controls the lounge volume with the vertical slider **Volume** control. This control is the **Advanced Slider** from the **Controls** palette in the Stardraw **Toolbox**, oriented vertically, and is named *LoungeVolumeControl*.

### Lounge Volume Level

#### Control Range and Display Scale

The HAL system uses a range of 0 to 1000 for level control values, corresponding to 0.0 to 100.0%. The Stardraw application's **Advanced Slider** must be configured to match this range in order to work correctly. To see this, click on the **Volume** slider control (named *LoungeVolumeControl*) and view the Properties for this control on the right side of the Stardraw window. In the Behavior section, the **Maximum** is 1000 while the **Minimum** is 0. These settings cause the slider control to have a range of 0 to 1000 when interacting with the HAL1 example driver, matching HAL system level control behavior.

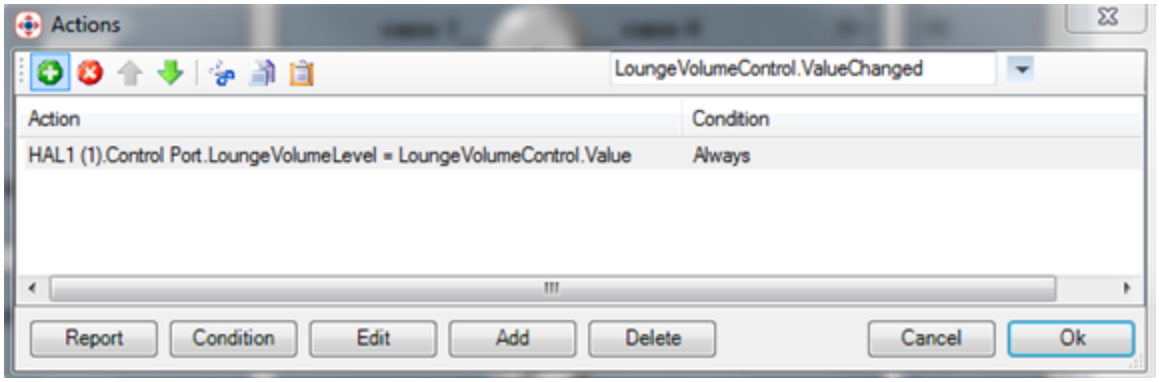
Changing the **Maximum** to 1000 also affects the slider's scale that the end user sees on the Stardraw form. In addition, the **Advanced Slider** control provides warning and danger zones, that color the scale for those zones brown and red respectively. For this example, we want a simple scale for the slider, displaying a range from 0 to 100 in ten steps and in a uniform black color. To see how the application achieves this, look at the *LoungeVolumeControl*'s Properties again and click + symbol next to the Labels property in the **Appearance** section. This property contains the set of labels that the control displays for its visible scale. We've also changed the **LabelsCount** to 11, matching the number of strings in the Labels array. To keep the color uniform black, we've also set the **DangerValue** and **WarningValue** properties to 1001, so they fall outside the range of the control.

Properties

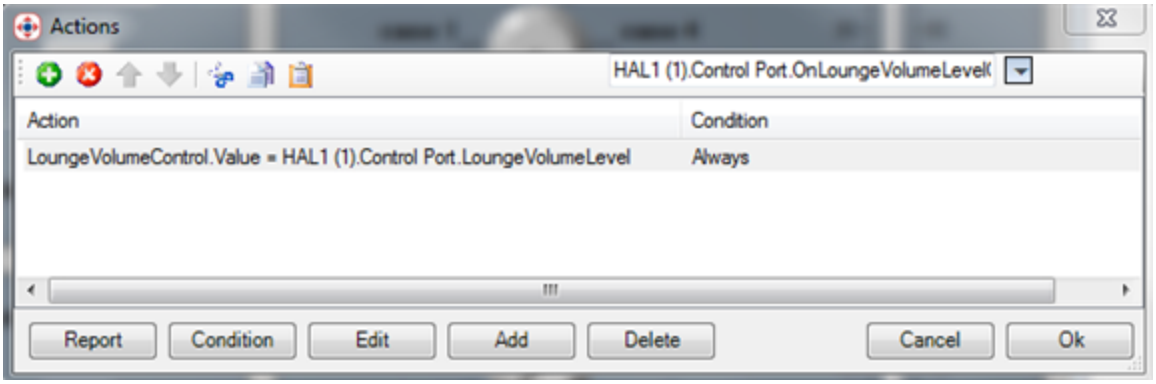
| | |
|---|---|
| InactiveBarColor | Gainsboro |
| InactiveLedColor | **MidnightBlue** |
| IndicatorWidth | 20 |
| ⊟ Labels | **String[] Array** |
| [0] | 0 |
| [1] | 10 |
| [2] | 20 |
| [3] | 30 |
| [4] | 40 |
| [5] | 50 |
| [6] | 60 |
| [7] | 70 |
| [8] | 80 |
| [9] | 90 |
| [10] | 100 |
| LabelsCount | **11** |
| LabelsPosition | Both |
| NumberScale | 1 |
| Orientation | **Vertical** |
| RightToLeft | No |
| ScaleColor | Black |
| ScaleColorMode | Sections |
| SliderHotImage | (none) |
| SliderImage | (none) |
| ⊞ SliderSize | **20, 20** |
| SliderStyle | Led |
| Text | **Lounge Volume** |
| TicksCount | 10 |
| TicksLenght | 8 |
| TicksPosition | Both |
| TicksSubDivisionsCour | 5 |
| UseWaitCursor | False |
| WarningColor | SaddleBrown |
| WarningValue | **1001** |
| ⊟ **Behavior** | |
| ContextMenuStrip | (none) |
| Enabled | True |
| ImeMode | NoControl |
| Increment | 10 |
| IndicatorOnly | False |
| Maximum | **1000** |
| Minimum | 0 |
| MinimumChangeEventP | 0 |
| ScaledValue | **0** |
| TabIndex | **0** |
| TabStop | True |
| Value | 0 |

**Current Level Value**

Now that you understand the **Advanced Slider**'s range and appearance, we can move on to the control events and associated actions. When the end user changes the volume level slider, Stardraw generates a **ValueChange** event. Each time this happens the application executes a specific action which sets the *LoungeVolumeLevel* property in the HAL1 driver to the current value of the form's *LoungeVolumeControl*. This in turn causes the driver to send a 'set level' message containing the new value to the Halogen/HAL Control Server, updating the HAL example configuration's *Lounge Level Control*. To see how the application associates the action with the event, double click on the slider control to bring up the **Actions** dialog for that control:




As with the toggle controls, the Halogen/HAL Control Server can also update the Stardraw application whenever the *Lounge Level Control* changes. Each time the level control in Halogen/HAL changes, the Control Server sends a 'set level' message containing the new value. When the Stardraw HAL1 driver receives this message, it generates an *OnLoungeVolumeLevelChanged* event. The example Stardraw application has this event mapped to an action that sets the *LoungeVolumeControl*'s **Value** to the driver's *LoungeVolumeLevel* property, which updates the value of the slider. To see this in the Actions dialog, use the drop down arrow for the event to navigate to Devices -> HAL1 (1) -> Control Port -> *OnLoungeVolumeChanged*. The Actions dialog that appears is:




## Selectors

The example application provides two selector controls, one for the lounge program source and one for the lounge tone. In the Stardraw form, both of these use the same built-in **MultiPositionKnob** control from the Stardraw **Toolbox Controls** palette.

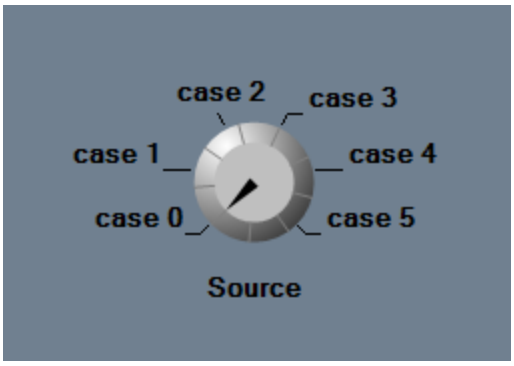
### Lounge Source Selector

The selector controls are the most complex controls in the Stardraw application and in the HAL1 example driver. This is because the application not only responds to changes to the current selection value, but it also dynamically

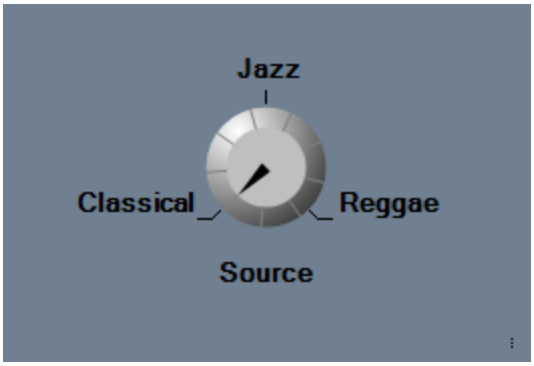
sets the number of selection items and the name of each item in response to messages that the Halogen/HAL Control Server sends.

#### Dynamic Selection Items

The lounge *Source Selector Control* in the Halogen example configuration contains three selections: Classical, Jazz, and Reggae. In the Stardraw form, however, the **MultiPositionKnob** control (named *SourceSelectionControl*) has the default set of selections and names:

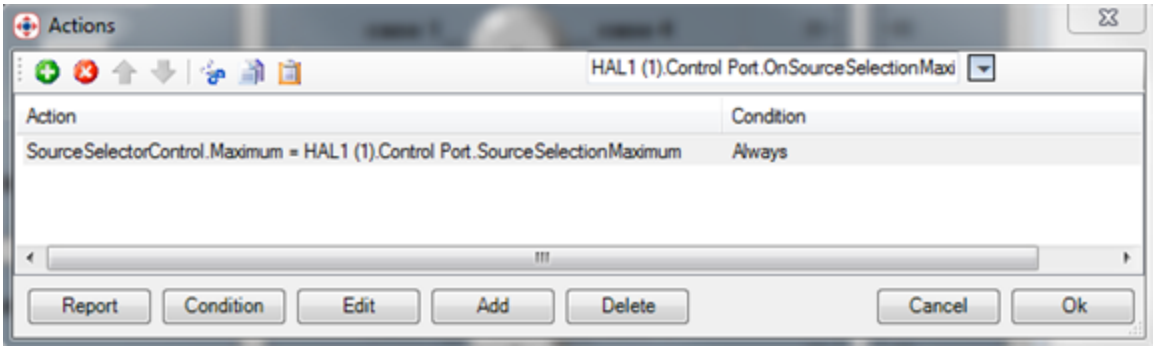



When you run the application, however, it looks like this:




"So," you ask, "what's going on here?" Well, the answer is that when the Stardraw application starts up, it asks the HAL1 driver for all of the current values of the controls defined for in the HAL configuration. This includes a request for the number of selection items in the *Source Selector Control*. When the Halogen/HAL Control Server returns the value (via a 'set selection link' message), the HAL1 driver notifies the Stardraw application so that it can set the number of items in *SourceSelectionControl* to match. In addition, whenever the HAL1 driver receives a 'set selection link' message, it then makes a request to the Halogen/Hal Control Server for each of the selection names. Halogen/HAL responds by sending a 'set selection name' message for each name, causing the Stardraw application to update its names in the *SourceSelectionControl*.

To see how the Stardraw application's *SourceSelectionControl* knob is configured to do this, double-click on the **Source MultiPositionKnob** control in the Stardraw form. In the Actions dialog that appears, use the event drop down arrow to navigate to the Devices -> HAL1 (1) -> Control Port -> *OnSourceSelectionMaximumChanged* event. The **Actions** dialog shows the event and associated actions:

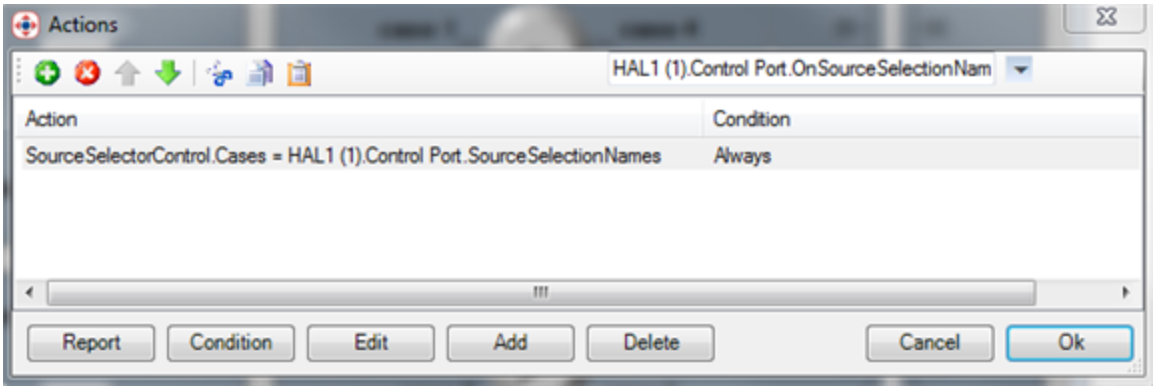



When the HAL1 Stardraw driver receives a 'set selection message' for the source selection control, it raises the *OnSourceSelectionMaximumChanged* event. The Stardraw action, shown above, sets the **Maximum** property of the *SourceSelectionControl* to the current value of the HAL1 driver's *SourceSelectionMaximum*. The *SourceSelectionControl*'s **Maximum** property determines the number of selection items for the **MultiPositionKnob** control.

### Dynamic Selection Names

The HAL1 driver includes a set of names for the Stardraw **Source** selector knob. Whenever the Halogen/HAL Control Server updates a selection name via a 'set selection name' message for the Source Selector Control, the driver records the new name and raises an *OnSourceSelectionNamesChanged* event. The Stardraw application has mapped an action to this event, which updates the set of names in the *SourceSelectionControl*. For the Stardraw **MultiPositionKnob**, the selection names are contained in the control's **Cases** property, a String array. When the *OnSourceSelectionNamesChanged* event occurs, the action sets the **Cases** property of *SourceSelectionControl* to the current value of the HAL1 driver's *SourceSelectionNames* driver. This one action takes care of setting all of the names at once.

To see this, use the drop down arrow in the **Actions** property for the Stardraw application's *SourceSelectionControl* to navigate to the Devices -> HAL1 (1) -> Control Port -> *OnSourceSelectionNamesChanged* event. The dialog shows the event and associated action as follows:

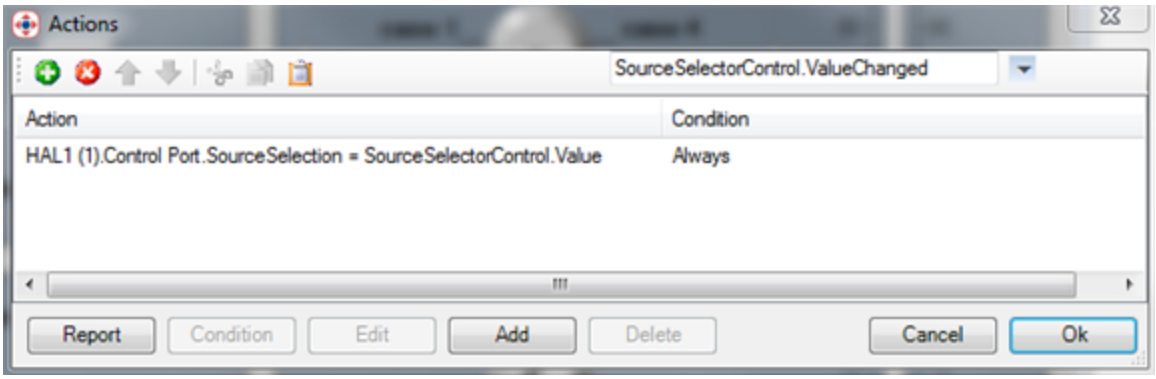



### Current Selection

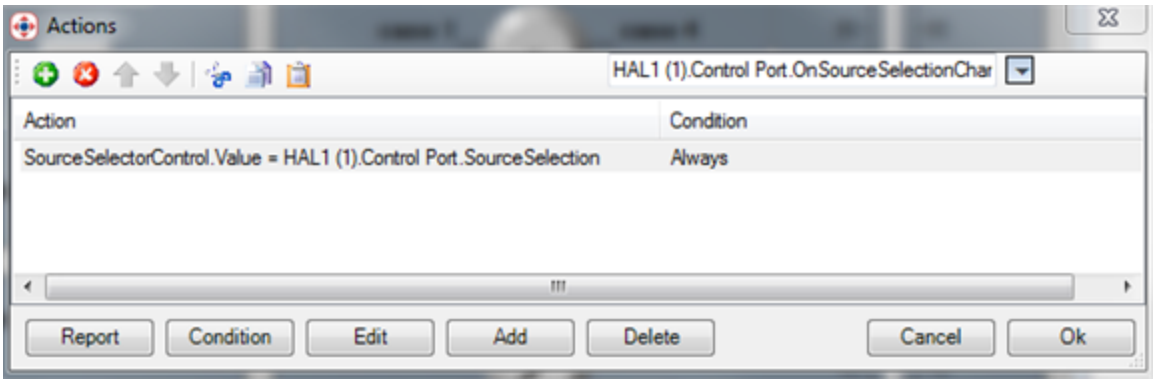
The value of the *SourceSelectionControl* is the currently selected item, which is a number starting at 0 for the first item and ranging to the maximum item, in this case 2 because there are a total of three selection items.

When the Stardraw end user changes the selector knob, the application generates a **ValueChanged** event for the *SourceSelectorControl*. The action for this event sets the HAL1 driver's *SourceSelection* property to the current **Value** of the Stardraw form's *SourceSelectorControl*. This in turn causes the driver to send a 'set selection'

message for the *Source Selector Control* to the Halogen/HAL Control Server. To see this, open the *SourceSelectorControl* by double clicking on the **Source** knob on the form. This brings up the **Actions** dialog for the *SourceSelectorControl*, which shows the **ValueChanged** event and the associated action:




The Stardraw example application also responds to *Source Selector Control* updates that the Halogen/HAL Control Server sends whenever the control changes in the HAL System. To do this, the application maps the HAL1 driver's *OnSourceSelectionChanged* event to an action that updates the form's *SourceSelectorControl* Value property. To see this, use the event drop down arrow in the Stardraw *SourceSelectorControl*'s **Actions** dialog to navigate to Devices -> HAL1 (1) -> Control Port -> *OnSourceSelectionChanged* event. The dialog changes to show the event and associated action:




## Lounge Tone Selector

The configuration of the Lounge tone selector **MultiPositionKnob** is very similar to the Source knob configuration. In this instance however, name of the form control is *ToneSelectorControl*, and the application maps the following events to actions:

| Description | Event | Action |
|---|---|---|
| Set the number of selector items in the MultiPostionKnob Tone control (ToneSelectorControl) | Devices -> HAL1 (1) -> Control Port -> *OnToneSelectionNamesChanged* | Set the *ToneSelectorControl* **Maximum** property to the HAL1 driver ToneSelectorMaximum property |
| Set the names of the ToneSelectorControl items | Devices -> HAL1 (1) -> Control Port -> *OnToneSelectionNamesChanged* | Set the *ToneSelectorControl* **Cases** property to the HAL1 drivers ToneSelectionNames property |

| Description | Event | Action |
|---|---|---|
| When the Stardraw end user changes the Tone knob setting. | Controls -> ToneSelectorControl -> **ValueChanged** | Set the HAL1 driver's *ToneSelection* property to the form's *ToneSelectionControl* **Value** property |
| When the Tone Selector Control changes in Halogen/HAL | Devices -> HAL1 (1) -> Control Port -> *OnToneSelectionChanged* | Set the *ToneSelectorControl* **Value** property to the HAL1 driver's *ToneSelection* property |

As before, you can see these events and associated actions by double clicking on the form's **Tone MultiPostionKnob** control and navigating to the event.

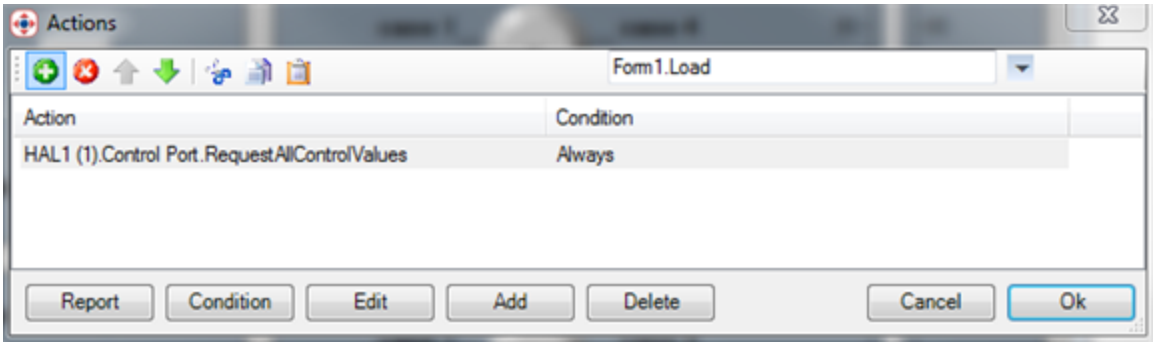
## Initializing the Form on Application Start

When the Stardraw example application starts, it is important to initialize all of the form controls to the current values of the external controls running in the HAL system. This will bring the Stardraw form to the same state as Halogen/HAL. To do this, the Stardraw application responds to a **Load** event from the *Form1* control. *Form1* is the name of the top level form of the example application and the **Load** event occurs right after the application begins running and connects to the Halogen/HAL Control Server.

The Stardraw application associates an action with the **Load** event, which calls the HAL1 driver's *RequestAllControlValues* method. In the HAL1 driver, this method sends a 'get' message to the Halogen/HAL Control Server for each of the external controls in the example HAL1 configuration.

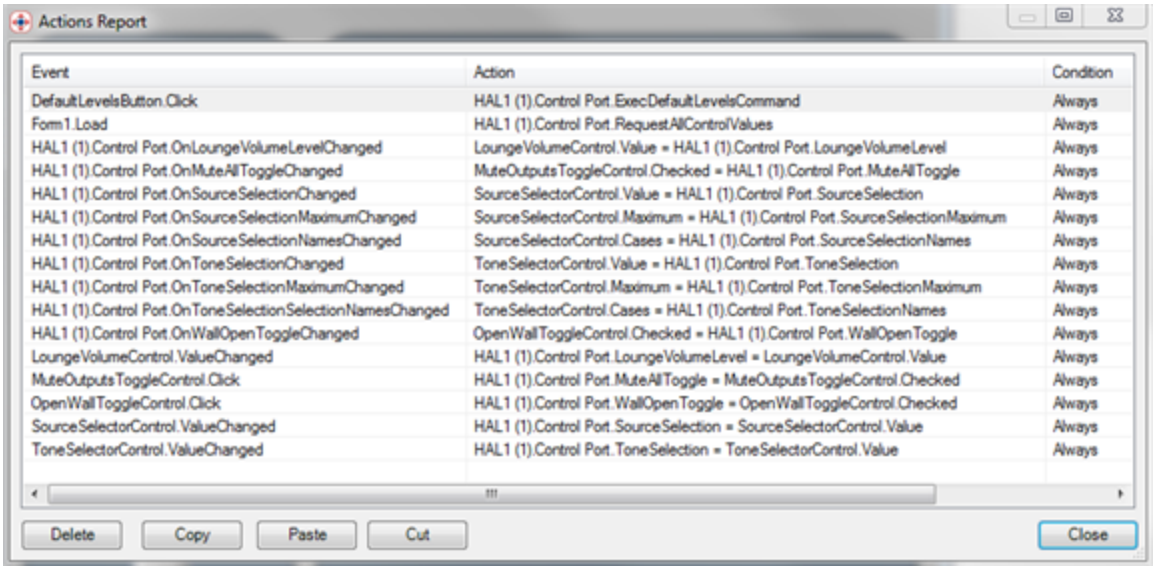
The Halogen/HAL Control Server responds with 'set' messages for each of the controls, which make their way through the Stardraw HAL1 driver, raising corresponding 'On…Changed' events, one for each control. After the Stardraw application performs the actions associated with these events, the form controls and the Halogen/HAL controls match.

To see how the application configures this event, bring up the **Actions** dialog for the Stardraw *Form1* by double clicking on the window title bar of the form. This is the light blue area across the top of the form that includes the Stardraw application icon in the top left of the form. The **Actions** dialog for *Form1* shows the *Form1*.**Load** event in the upper left and the associated action:




## Events/Actions Report

You can see all of the events and associated actions for the example Stardraw application by clicking on the **Report** button in the bottom left corner of the Stardraw **Action** dialog. Regardless of which event the **Actions** dialog is currently displaying, clicking on **Report** brings up a dialog that shows all of the configured events in the application. For the example application, this is as follows:

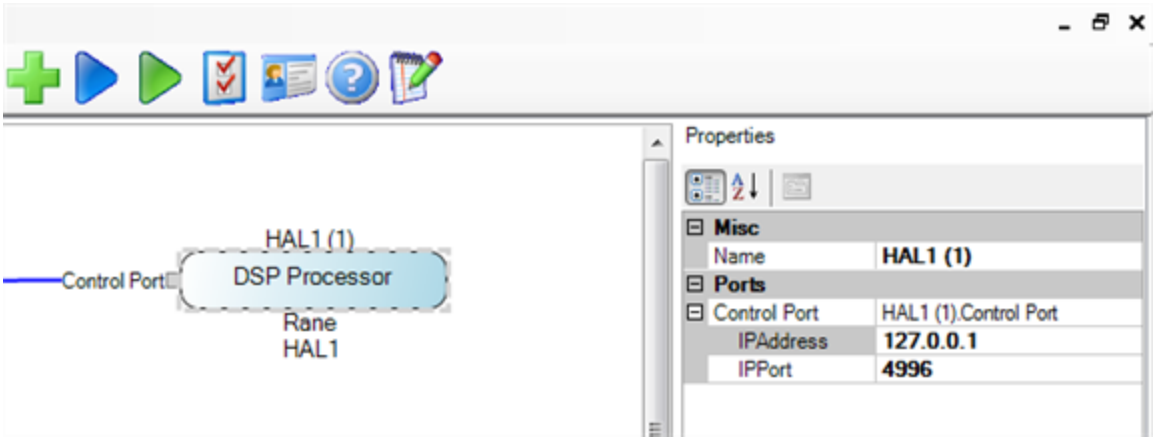



# Running and Testing the Application

## Connecting the Halogen Control Server

Initially, the example application is set up to use Halogen as the Control Server running on the same PC as the Stardraw authoring application. To accomplish this, it configures the HAL1 device in the Stardraw topology to use the PC's loopback IP address of 127.0.0.1. This IP address always refers to a virtual network address of the current computer. The word virtual in this case means that even though the address is a real IP address, the communications exchanged with a 'device' at this address is only routed internally within the PC. No network packets actually travel on a physical network.

To view and change the IP address for the HAL1, change to the **Topology** window in the Stardraw authoring application by clicking on the **Topology** tab in the lower left corner. Now, click on the HAL1 (1) DSP Processor block to view its properties. Click on the + symbol next to Control Port to view the settings for the port:




As you can see, the IPAddress is set to the loopback address.

If you want to run Halogen on one PC and the example Stardraw application on another, just change the IPAddress for the HAL1 (1) to the address of the PC that Halogen is running on.

## Running the Application

First, be sure that Halogen is running and that the example configuration loaded. To do this:

- Double click on the Halogen Icon on your Windows desktop.
- After Halogen starts, open the example configuration file by clicking the **Open** button at the top of the Halogen window.
- In the Open Halogen Configuration dialog that appears, navigate to the Halogen Stardraw Example directory and select the ControlSystemSample.hal file.

Now run the Stardraw example application from the Stardraw authoring application by clicking on the blue (Run Project) arrow in the Stardraw toolbar.

The example form will appear and the controls will match the corresponding external controls in Halogen.

In Halogen, switch to the Processing Workspace by clicking on the **Processing** tab near the top right of the Halogen application window. Now, click on the **Control Systems** button then **External Controls** to open the External Control Systems dialog. Click on each of the tabs, **Level**, **Select**, **Toggle**, and **Command** to see the configured external controls. To test the controls, change them in either the Stardraw application or in Halogen and observe the behavior of both applications.

For example, to test the lounge volume level control:

- Open the Control Systems dialog on Halogen and click on the Level tab.
- Click on the hyperlink for the control in the Halogen Control System dialog, Lounge Level Control.
- Open the *HAL1x Line Output (1)* block in the top right area of the configuration's Processing Map by double clicking on the **HAL1x:Line** title bar region of the block.

**NOTE:** Note: The three level controls (The Stardraw Volume slider, the Halogen Lounge Level Control and the HAL1x Line Output (1) level control) are connected to each other and always track together.

- Change any of the three level controls and observe the changes that occur in the Stardraw application and in Halogen. A change made to one level causes the other two to change to the same relative setting:

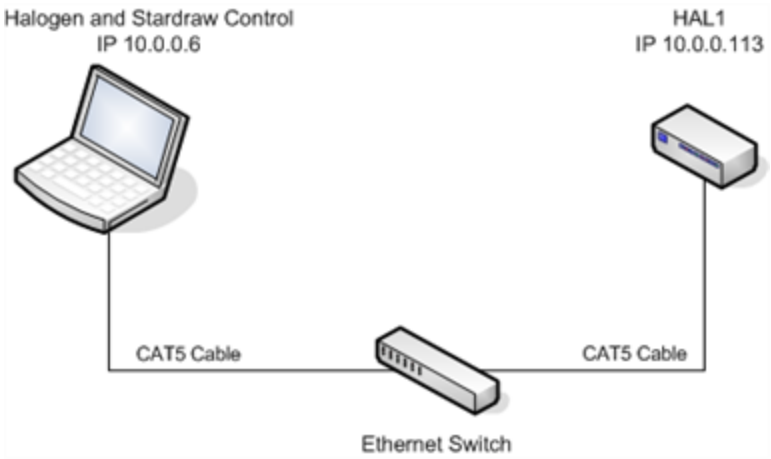
You can try this out for all of the other controls in the Stardraw example application and the corresponding controls in the Halogen example configuration. Here is a complete list of controls in the example and how they relate to each other:

| Stardraw Application | Halogen Application | |
|---|---|---|
| Control Name (Control Properties) | External Control (Control Systems Dialog) | Operational Control (Linked to External Control) |
| LoungeVolumeControl | Lounge Level Control | Line Output (1) - Level |
| SourceSelectorControl | Source Selector Control | Selector (1) - Selector |

| Stardraw Application | Halogen Application | |
|---|---|---|
| ToneSelectorControl | PEQ Selector Control | PEQ Preset Selector - Selector |
| OpenWallToggleControl | Wall Toggle Control | Wall (1) – Open Toggle |
| MuteOutputsToggleControl | Mute All Toggle Control | Mute All Preset – Active Toggle |
| DefaultLevelsButton | Default Command Control | Default Level Preset – Assert Command |

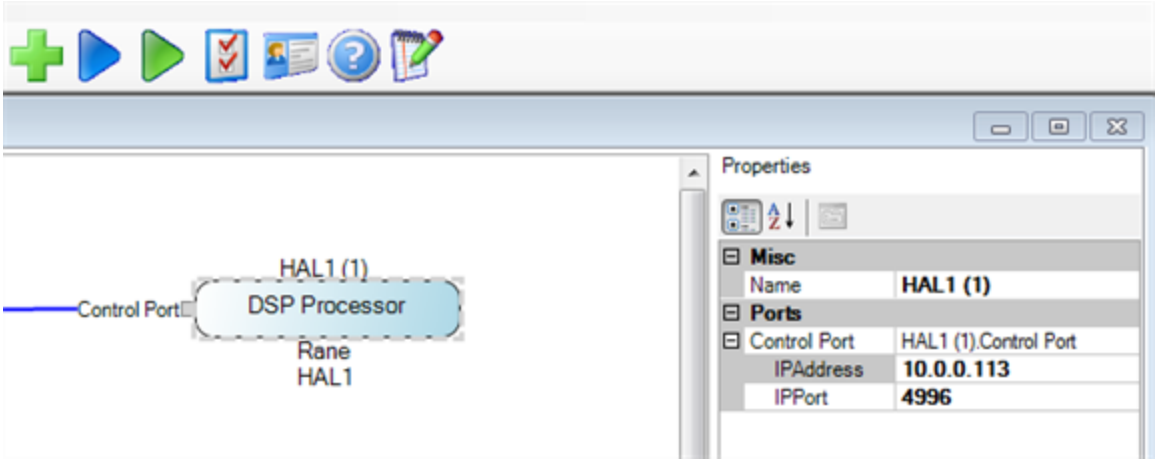
### Running the Application with a HAL1x

Once you have tried out the example Stardraw application with Halogen and have a HAL1x available, you can switch to using the HAL1x as the Control Server. The simplest way to do this is to connect your PC and your HAL1x to the same network using an Ethernet switch. Minimally, the components of this network are:




The IP addresses shown are examples, but it is important that the PC and the HAL1x have compatible addresses. We suggest that you configure a static IP address on the HAL1x that is compatible with the PC's network. That is, both the PC and the HAL1x are on the same subnet. For example, if your PC has an IP address of 10.0.0.6 and a subnet mask of 255.255.255.0, you could configure your HAL1x to have a static IP address of 10.0.0.129, with a subnet mask of 255.255.255.0. For information on how to set HAL1x's static IP address and subnet mask see *To connect to the HAL external control server...* in Appendix B: "Using PuTTY to Test External Control Systems" on page 51.

Once your PC and HAL1x are on the same network, you need to configure the Stardraw application to use HAL1x's IP address. To do this, change to the **Topology** window in the Stardraw authoring application by clicking on the **Topology** tab in the lower left corner. Now, click on the HAL1 (1) DSP Processor block to view its properties. Click on the + symbol next to Control Port to view the settings for the port and change the IPAddress to the address of your HAL1. In following example it is set to 10.0.0.113:




To get everything running:

- In Halogen connect to your HAL1x. Do this by clicking the **Connect** button in Halogen, then clicking on the **Connect** button in the **Connect to Device** window that appears.
- Load the example HAL1x configuration. Do this by clicking on Halogen's **Load** button and selecting the example configuration file, ControlSystemSample.hal, included in the Stardraw Control support package.
- Start the Stardraw Control example application by clicking on the blue (Run Project) arrow in the Stardraw authoring application.

Now the Stardraw application is connected to the Control Server on your HAL1x. If you operate the controls on the Stardraw application panel, for example, the Stardraw HAL1 driver sends messages to the HAL Control Server, which change the corresponding controls in your HAL1x.

At this point you could disconnect Halogen from the HAL1x or even close Halogen because the Stardraw application communicates directly with the HAL1x and Halogen is no longer required as part of the control system.

## Troubleshooting

1. If the Stardraw application cannot connect to the Halogen Control Server:
   - Make sure Halogen is running and you have opened the sample configuration (ControlSystemSample.hal)
   - Look at the Stardraw control authoring application's Debug window. This window automatically appears when you run the Stardraw application. The debug information will help you learn why it can't connect to Halogen.
   - If Halogen and Stardraw are running in the same PC, make sure the IP address configured for the HAL1 in the Stardraw application is the local loopback address 127.0.0.1 and the port is 4996.
   - If Halogen and Stardraw are running on different PCs:
     - Make sure both PCs are on the same network and have compatible IP addresses.
     - Make sure the IP address configured for the HAL1 in the Stardraw application is the IP address of the PC running Halogen and that the port is 4996.
     - Ensure that the firewalls on both PC's allow TCP/IP communication on port 4996.
2. If the Stardraw application connects to the Halogen Control Server but the controls do not operate as

expected:

- Make sure you have opened the example configuration in Halogen (ControlSystemSample.hal)

3. If Stardraw can't connect to the HAL Control Server:
   - Make sure the PC running Stardraw and Halogen are on the same network as the HAL1.
   - Ensure that the PC and HAL1x have compatible IP addresses.
   - Use a static IP on HAL1x and make sure it doesn't conflict with any other IP address on the network, including those automatically assigned by a DHCP server.
   - Check the IP address configured in Stardraw application HAL1 properties. Make sure it is the HAL1x's IP address and the port is 4996.
   - Ensure that the firewall on the PC allows TCP/IP communication on port 4996.
   - Look at the Stardraw control authoring application's Debug window. This window automatically appears when you run the Stardraw application. The debug information will help you learn why it can't connect to Halogen.
4. If the Stardraw application connects to HAL1x but the controls do not operate as expected:
   - Make sure you have applied the configuration to the HAL1x (ControlSystemSample.hal). You can also do this by connecting to your HAL1x then loading the example configuration.
   - Use a telnet client to monitor and exercise the external control messages. See Appendix B: "Using PuTTY to Test External Control Systems" on page 51.

# APPENDIX A: HAL External Control Message Protocol

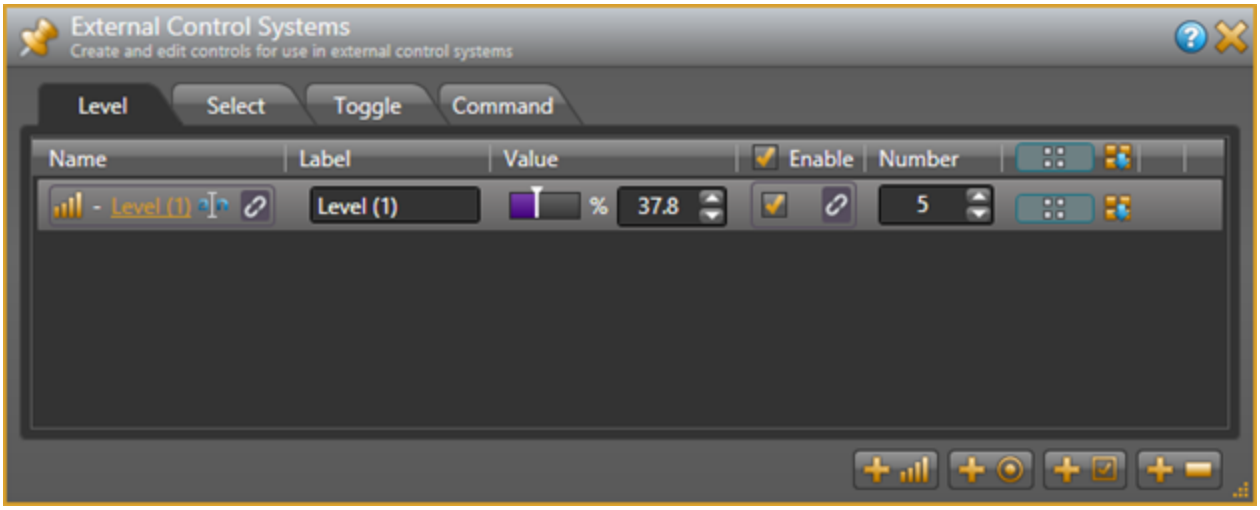
Once you have connected to either the Halogen or HAL external control server, you can communicate with it using simple, text-based messages. This section defines this message protocol.

All messages consist of ASCII text and a single message always starts with a '<' character and ends with a '>' character. Inside the start and end characters is the message body, which always includes at least a message type followed by a control number. Each part of the message body is separated from another part by an '`&`' character. Messages can also have additional parts depending on the type of message. The table below shows the format for each specific message type.

When sending messages to the Halogen/HAL Control Server, it:

- Allows white space anywhere in the message, which includes space, tab, carriage return, and line feed characters
- Ignores all characters outside of a message (that is, before the start character and after the end character)
- Ignores the case of alpha characters – you can use upper or lower case in any combination

Halogen's External Control Systems dialog defines the unique control number for each control, which is always '**n**' in the message table below. For example, suppose you have defined a Level control in the Halogen External Control Systems dialog as follows:




In this instance, the control number is 5, and that is what you use in the 'set level' and 'get level' messages when developing external control systems. In this case the 'get level' message for this control is `<L&5>`.

## Details of external control messages

In the following tables, normal characters are literal while bold characters are variable and defined in the description.

## All Controls

| Message Format | Description | How Used | Example |
|---|---|---|---|
| <?> | Get all values message. | Send this to HAL to request the current values of all controls. | <?> - request the current values of all controls in the configuration. |

## Level Controls

| Message Format | Description | How Used | Example |
|---|---|---|---|
| <?&L> | Get all level control values message.<br>The includes value, enable, and label for every level control. | Send this to HAL to request all values associated with **all** level controls. | <?&L> - request the current level value, enable, and label of all Level external controls. |
| <?&L&**n**> | Get level value, enable and label message, where:<br>**n** is the control number. | Send this to HAL to request all values associated with the level control with the specified control number. | <?&L&1> - request the current value, enable, and label of Level external control number 1. |
| <L&**n**> | Get level message, where:<br>**n** is the control number. | Send this to HAL to request the current value of a Level external control. | <L&1> - request the current value of Level external control number 1. |
| <L&**n**&**v**> | Set level message, where:<br>**n** is the control number.<br>**v** is the control value.<br>The range for v is 0 to 1000, corresponding to 0 to 100.0%. | Send this to HAL to set a Level external control to a new value.<br>HAL sends this:<br>1. in response to a get level message<br>2. whenever a Level external control changes value | <L&1&456> - set the value of Level external control number 1 to 45.6%. |

| Message Format | Description | How Used | Example |
|---|---|---|---|
| <L&**n**&+**v**><br><br><L&**n**&-**v**> | Increment/Decrement level message, where:<br><br>**n** is the control number.<br><br>+ indicates increment.<br><br>- indicates decrement.<br><br>**v** is the amount to increment or decrement the current control value.<br><br>The range for v is 0 to 1000, corresponding to 0 to 100.0%.<br><br>**NOTE:** When an increment or decrement message would result in setting the level control to a value outside of the allowed range of 0 to 100%, the HAL sets the level to the limit value. | Send this to HAL to increment or decrement a Level external control. | <L&1&+105> - increment the value of Level external control number 1 by 10.5%.<br><br><L&2&-234> - decrement the value of Level external control number 2 by 23.4%. |
| <LA&**n**> | Get label message, where:<br><br>**n** is the control number. | Send this to HAL to request the current label value of a Level external control. | <LA&1> - request the current label value of Level external control number 1. |
| <LA&**n**&**string**> | Set label message, where:<br><br>**n** is the control number.<br><br>**string** is the label for the control. | Send this to HAL to set the label for a Level external control to a new value.<br><br>HAL sends this:<br><br>1. in response to a get label message<br>2. whenever a label for a Level external control changes value | <LA&1&-Volume> - set the label value for Level control number 1 to "Volume". |

| Message Format | Description | How Used | Example |
|---|---|---|---|
| <LE&**n**> | Get enable message, where:<br>**n** is the control number. | Send this to HAL to request the current enable value of a Level external control. | <LE&4> - request the current enable value of Level external control number 4. |
| <LE&**n**&**v**> | Set enable message, where:<br>**n** is the control number.<br>**v** is the enable value.<br>The values for v are 0 (disabled) or 1 (enabled). | Send this to HAL to set the enable for a Level external control to a new value.<br>HAL sends this:<br>1. in response to a get enable message<br>2. whenever an enable for a Level external control changes value | <LA&4&0> - set the enable value for Level control number 4 to 0 (disabled). |

## Toggle Controls

| Message Format | Description | How Used | Example |
|---|---|---|---|
| <?&T> | Get all toggle control values message.<br>The includes value, enable, and label for every toggle control. | Send this to HAL to request all values associated with **all** toggle controls. | <?&T> - request the current toggle value, enable, and label of all Toggle external controls. |
| <?&T&**n**> | Get toggle value, enable and label message, where:<br>**n** is the control number. | Send this to HAL to request all values associated with the toggle control with the specified control number. | <?&T&1> - request the current value, enable, and label of Toggle external control number 1. |
| <T&**n**> | Get toggle message, where:<br>**n** is the control number. | Send this to HAL to request the current value of a Toggle external control. | <T&23> - request the current value of Toggle external control number 23. |

| Message Format | Description | How Used | Example |
|---|---|---|---|
| <T&**n**&**v**> | Set toggle message, where:<br>**n** is the control number.<br>**v** is the control value.<br>The values for v are 0 (unchecked) or 1 (checked). | Send this to HAL to set a Toggle external control to a new value.<br>HAL sends this:<br>1. in response to a get toggle message<br>2. whenever a Toggle external control changes value | <T&23&1> - set the value of Toggle external control number 23 to 1 (checked). |
| <TA&**n**> | Get label message, where:<br>**n** is the control number. | Send this to HAL to request the current label value of a Toggle external control. | <TA&1> - request the current label value of Toggle external control number 1. |
| <TA&**n**&**string**> | Set label message, where:<br>**n** is the control number.<br>**string** is the label for the control. | Send this to HAL to set the label for a Toggle external control to a new value.<br>HAL sends this:<br>1. in response to a get label message<br>2. whenever a label for a Toggle external control changes value | <TA&1&Mute> - set the label value for Toggle control number 1 to "Mute". |

| Message Format | Description | How Used | Example |
|---|---|---|---|
| <TE&**n**> | Get enable message, where:<br>**n** is the control number. | Send this to HAL to request the current enable value of a Toggle external control. | <TE&4> - request the current enable value of Toggle external control number 4. |
| <TE&**n**&**v**> | Set enable message, where:<br>**n** is the control number.<br>**v** is the enable value.<br>The values for v are 0 (disabled) or 1 (enabled). | Send this to HAL to set the enable for a Toggle external control to a new value.<br>HAL sends this:<br>1. in response to a get enable message<br>2. whenever an enable for a Toggle external control changes value | <TA&4&0> - set the enable value for Toggle control number 4 to 0 (disabled). |

## Selector Controls

| Message Format | Description | How Used | Example |
|---|---|---|---|
| <?&S> | Get all selector control values message.<br>The includes selector count, value, enable, label and item names for every selector control. | Send this to HAL to request all values associated with **all**selector controls. | <?&S> - request the current selector count, value, enable, label and item names of all Selector external controls. |
| <?&S&**n**> | Get selector count, value, enable, label and item names message, where:<br>**n** is the control number. | Send this to HAL to request all values associated with the selector control with the specified control number. | <?&S&1> - request the current selector count, value, enable, label and item names of Selector external control number 1. |
| <S&**n**> | Get selection message, where:<br>**n** is the control number. | Send this to HAL to request the current value of a Select external control. | <S&6> - request the current value of Select external control number 6. |

| Message Format | Description | How Used | Example |
|---|---|---|---|
| <S&**n**&**v**> | Set selection message, where:<br>**n** is the control number.<br>**v** is the control value.<br>The range for v is 0 to one less than the number of selections. | Send this to HAL to set a Select external control to a new value.<br>HAL sends this:<br>1. in response to a get selection message<br>2. whenever a Select external control changes value | <S&6&3> - set the value of Select external control number 6 to 3 (the fourth item in the set of selections). |
| <SL&**n**> | Get selection link message, where:<br>**n** is the control number. | Send this to HAL to request the current number of selections for Select external control. | <SL&7> - get the number of selections for Select external control number 7. |
| <SL&**n**&**x**> | Set selection link message, where:<br>**n** is the control number.<br>**x** is the number of selections. 0 indicates that no selections are available – i.e. the selector link is inactive. | It is not meaningful to send this message to HAL because an external control system cannot change a selection link.<br>HAL sends this:<br>1. in response to a get selection link message<br>2. whenever the link activation changes for the Select external control | <SL&7&4> - set the number of selections for Select external control number 7 to 4. |
| <SN&**n**&**i**> | Get selection name message, where:<br>1. **n** is the control number.<br>2. **i** is the selection item. The range for i is 0 to one less than the number of selections. | Send this to HAL to request the current value of a selection name for a particular selection item of a Select external control. | <SN&12&2> - get the current selection name for selection item 2 of Select control number 12.<br>Selection item 2 is the third item in the set of selections. |

| Message Format | Description | How Used | Example |
|---|---|---|---|
| <SN&**n**&**v**&**string**> | Set selection name message, where:<br>**n** is the control number.<br>**v** is the control value. The range for v is 0 to one less than the number of selections.<br>**string** is the current display name for selection **v**. | It is not meaningful to send this message to HAL because an external control system cannot change a selection name.<br>HAL sends this:<br>1. in response to a get selection name message<br>2. whenever the link activation changes for the Select external control<br>3. whenever one of the selection names change | <SN&12&2&Jazz> - set the current selection name for selection item 2 of Select control number 12 to “Jazz”.<br>Selection item 2 is the third item in the set of selections. |
| <SA&**n**> | Get label message, where:<br>**n** is the control number. | Send this to HAL to request the current label value of a Selector external control. | <SA&1> - request the current label value of Selector external control number 1. |
| <SA&**n**&**string**> | Set label message, where:<br>**n** is the control number.<br>**string** is the label for the control. | Send this to HAL to set the label for a Selector external control to a new value.<br>HAL sends this:<br>1. in response to a get label message<br>2. whenever a label for a Selector external control changes value | <SA&1&Program> - set the label value for Selector control number 1 to "Program". |

| Message Format | Description | How Used | Example |
|---|---|---|---|
| <SE&**n**> | Get enable message, where:<br>**n** is the control number. | Send this to HAL to request the current enable value of a Level external control. | <SE&4> - request the current enable value of Selector external control number 4. |
| <SE&**n**&**v**> | Set enable message, where:<br>**n** is the control number.<br>**v** is the enable value.<br>The values for v are 0 (disabled) or 1 (enabled). | Send this to HAL to set the enable for a Selector external control to a new value.<br>HAL sends this:<br>1. in response to a get enable message<br>2. whenever an enable for a Selector external control changes value | <SA&4&0> - set the enable value for Selector control number 4 to 0 (disabled). |

## Command Controls

| Message Format | Description | How Used | Example |
|---|---|---|---|
| <?&C> | Get all command control labels message. | Send this to HAL to request label value associated with **all** command controls. | <?&C> - request the current label of all Command external controls. |
| <?&C&**n**> | Get the label message, where:<br>**n** is the control number.<br>NOTE: Equivalent to the <CA&**n**> message described below. | Send this to HAL to request the current label value of a Command external control with the specified control number. | <?&C&1> - request the current label of Command external control number 1. |
| <C&**n**> | Command message, where:<br>**n** is the control number. | Send this to HAL to fire a Command external control.<br>HAL sends this whenever a Command external control fires. | <C&4> - Fire Command control number 4. |

| Message Format | Description | How Used | Example |
|---|---|---|---|
| <CA&**n**> | Get label message, where:<br>**n** is the control number. | Send this to HAL to request the current label value of a Command external control. | <CA&1> - request the current label value of Command external control number 1. |
| <CA&**n**&**string**> | Set label message, where:<br>**n** is the control number.<br>**string** is the label for the control. | Send this to HAL to set the label for a Command external control to a new value.<br>HAL sends this:<br>1. in response to a get label message<br>2. whenever a label for a Command external control changes value | <CA&1&Flatten> - set the label value for Command control number 1 to "Flatten". |
| <CE&**n**> | Get enable message, where:<br>**n** is the control number. | Send this to HAL to request the current enable value of a Command external control. | <CE&4> - request the current enable value of Command external control number 4. |
| <CE&**n**&**v**> | Set enable message, where:<br>**n** is the control number.<br>**v** is the enable value.<br>The values for v are 0 (disabled) or 1 (enabled). | Send this to HAL to set the enable for a Command external control to a new value.<br>HAL sends this:<br>1. in response to a get enable message<br>2. whenever an enable for a Command external control changes value | <CA&4&0> - set the enable value for Command control number 4 to 0 (disabled). |

## Communications Monitoring

| Message Format | Description | How Used | Example |
|---|---|---|---|
| <PING> | Ping message | Send this to HAL to request a pong message. | <PING> |
| <PONG> | Pong message | It is not meaningful to send this message to HAL.<br>HAL sends this whenever it receives a ping message from an external control system. | <PONG> |

# APPENDIX B: Using PuTTY to Test External Control Systems

The Halogen/HAL external control message protocol is ASCII text based, which means that you can use a standard telnet client to connect to the control server and exchange messages. This is often useful when developing and testing a HAL configuration that contains external controls, either to monitor the operation of your external control system or to serve as a simple form of external control system before the production system is ready or available.

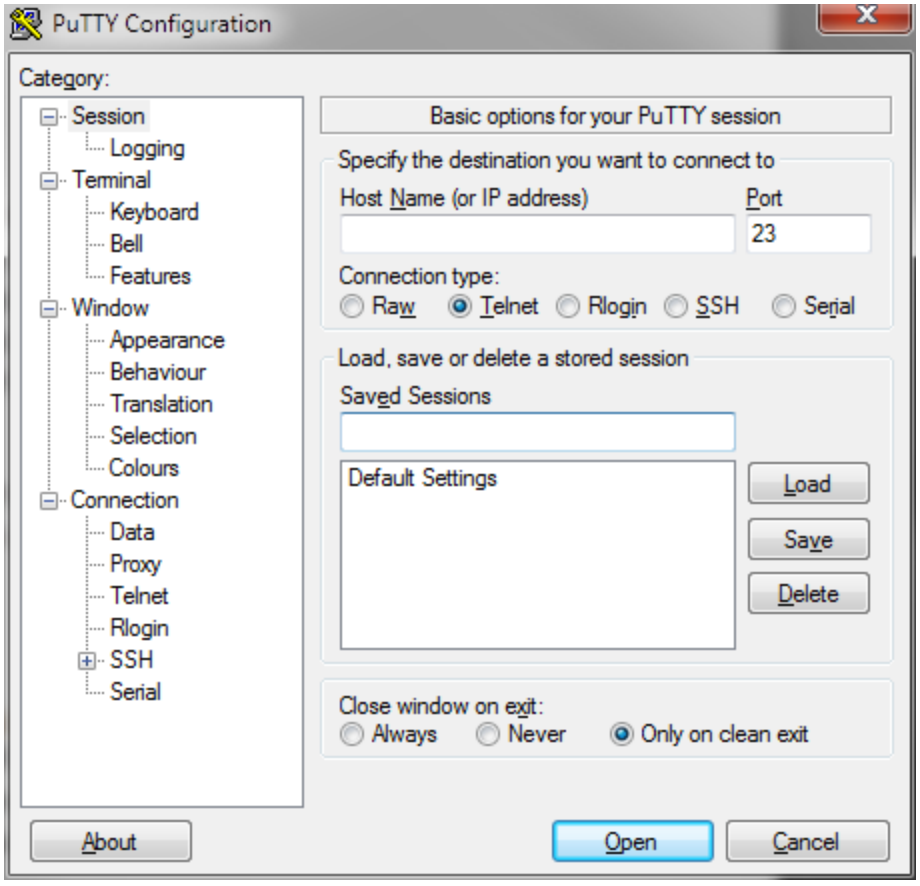
Any telnet client will work, but this section shows how to do this with PuTTY, an open source telnet client application that is free, easy to install and use, and runs on all versions of Windows.

### How to obtain and install PuTTY

The top level web page for PuTTY is http://www.putty.org. That page contains a link to a page that has links for downloading the PuTTY application. All you need is the latest release version of putty.exe for Windows on Intel x86. Download this file and put it in an accessible folder on your PC. Since the complete installation is a single file, putting it on the desktop works great.

### Starting PuTTY and configuring it for communications with the Halogen control server

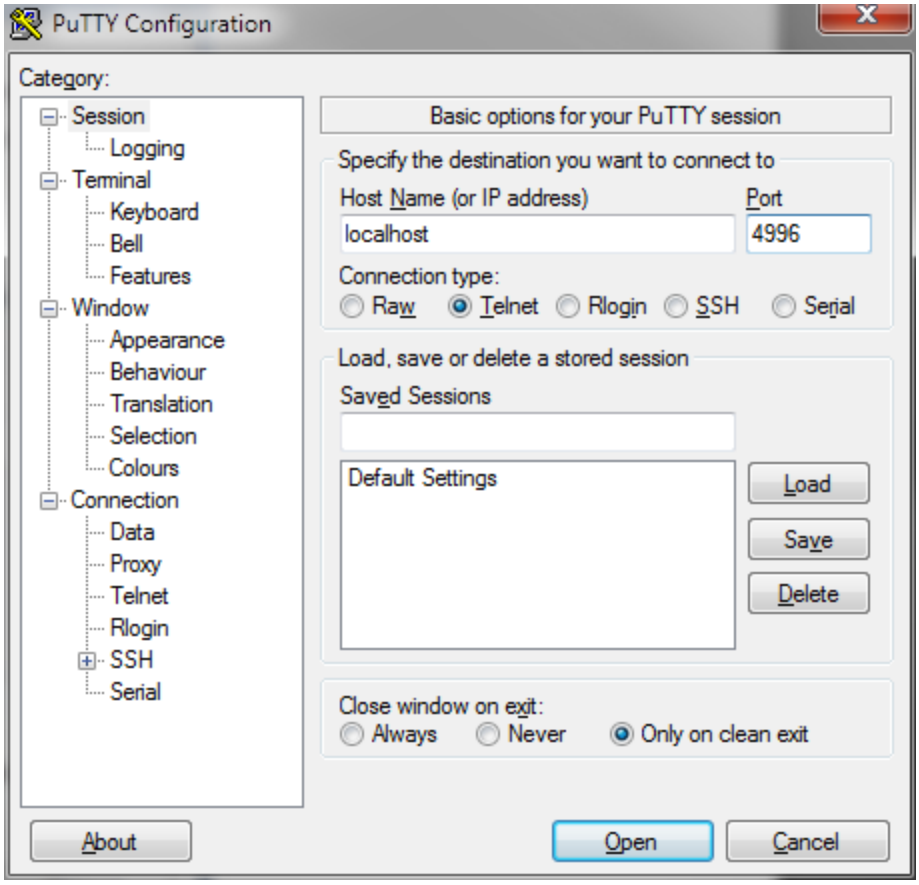
Start PuTTY by double clicking on the putty.exe file. This brings up the PuTTY Configuration window:




### To connect to the Halogen external control server, do the following:

1. Start Halogen and load your configuration that contains the external controls you'd like to test

2. In the PuTTY configuration window:
   a. Enter localhost in the Host Name text box
   b. Change the Connection type to Telnet
   c. Change the port number to 4996
   d. Click the Open button

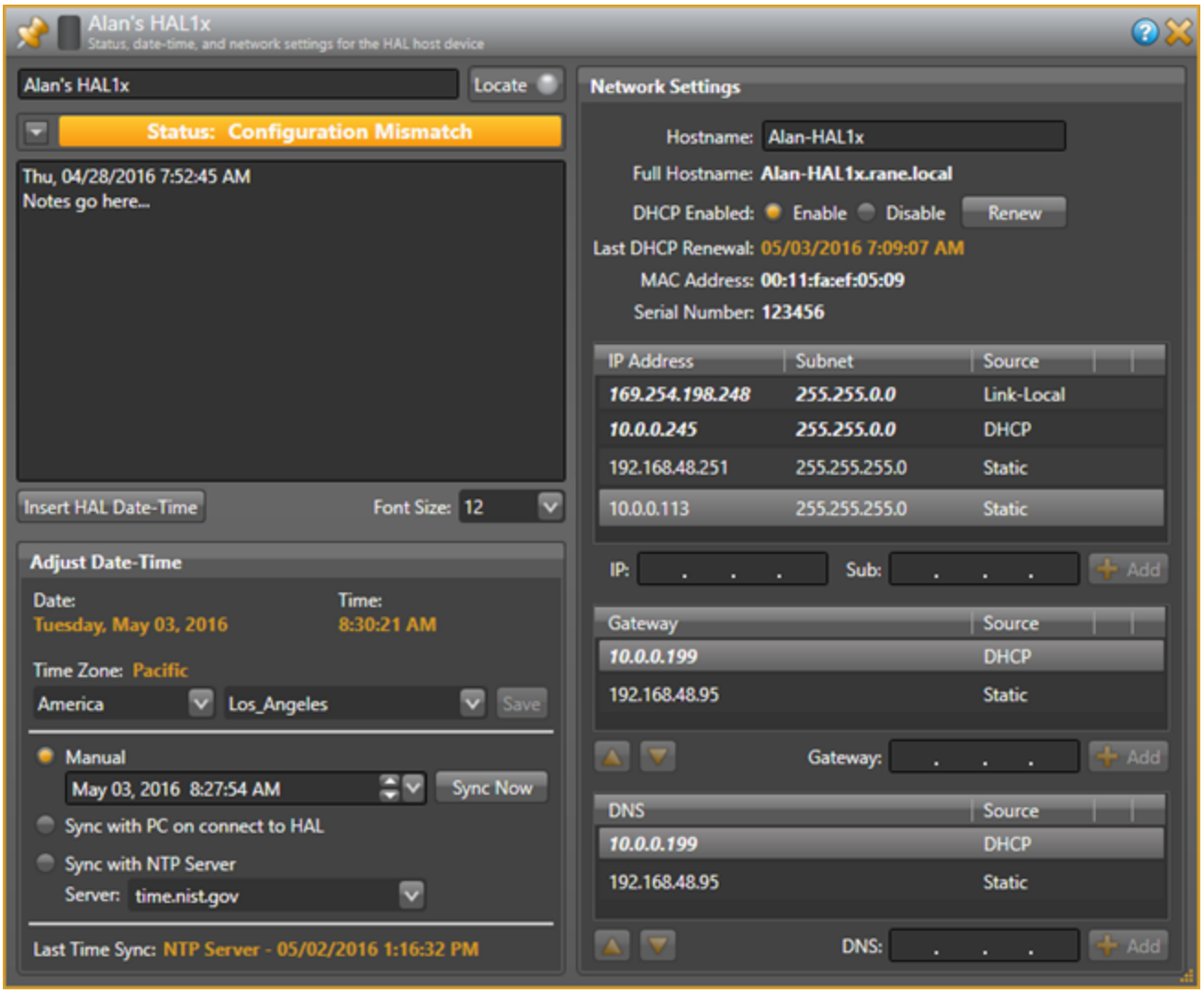



**To connect to the HAL external control server, do the following:**

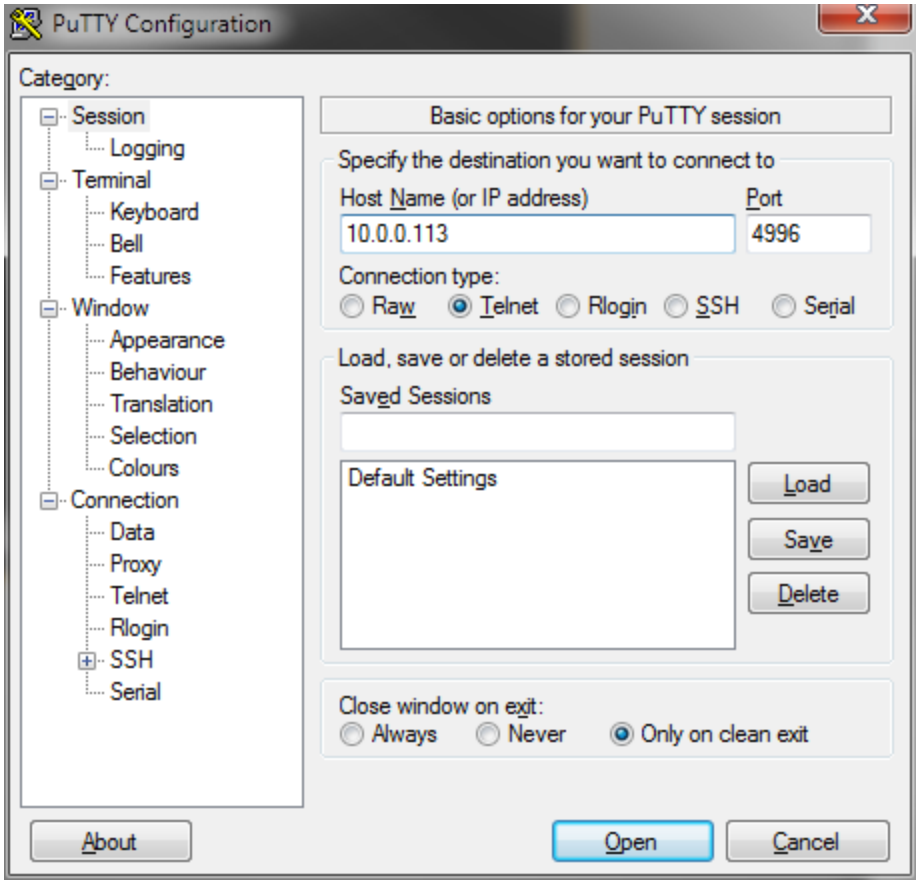
1. Start Halogen and Apply your configuration that contains the external controls you'd like to test to the HAL.
2. Open the HAL property dialog by double clicking on the HAL title bar in the Halogen Hardware Workspace.

3. Observe the HAL's configured IP addresses and pick one that you want to use with PuTTY. The address you choose needs to be on the same network as your PC that is running PuTTY.
4. In the PuTTY configuration window:
   a. Enter HAL's IP address in the Host Name text box
   b. Change the Connection type to Telnet
   c. Change the port number to 4996
   d. Click the Open button

For example, the HAL1 host properties dialog will look something like this:



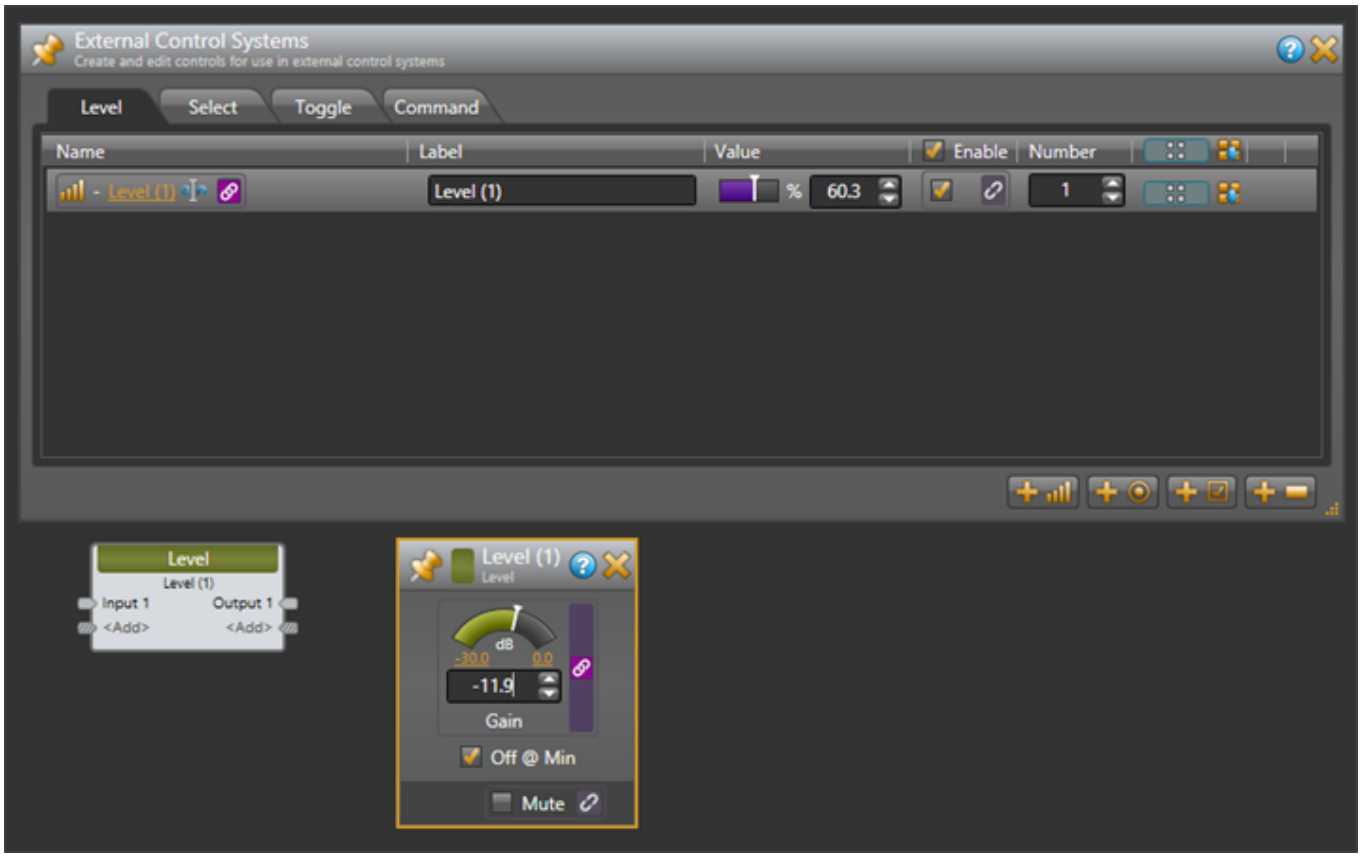

Here, We've added a static IP of 10.0.0.113. We'll use this in the PuTTY Configuration window:




## Communicating with the Halogen/HAL control server

Connecting PuTTY to either Halogen or HAL establishes communication between the telnet client and the control server. Now you can send and receive messages using the PuTTY window that appears. For example, suppose you have a Level control configured in Halogen that is control number 1 and currently has a value of 60.5%. To request the current value of this control, type <L&1> into the PuTTY window and then press Enter. This will send the message to Halogen, which will respond with <L&1&605>. If all is right with your test system, you'll see something like this:

Your Halogen configuration will be something like this: (the Level external control is linked to the Level block's level control)




## How to use PuTTY when testing external control systems

Here are a few ideas for what you can do:

1. Test your HAL external controls by sending and receiving external control messages

   To do this, connect PuTTY to your HAL or Halogen external control server and type external control messages into the PuTTY window and observe what HAL sends in response.

2. See how the HAL control server responds when you change external controls either from Halogen or from actual remotes in the HAL system.

   Connect PuTTY to your HAL or Halogen external control server and change external controls from within the HAL system. Observe what the control server sends to PuTTY.

3. See how the HAL control server responds when configured presets activate or deactivate that affect external control values or link activations.

   Connect PuTTY to your HAL or Halogen external control server and activate or deactivate presets that contain blocks or links related to external controls. Observe what the control server sends to PuTTY.

4. Use PuTTY to monitor the operation of your external control system

You can connect any number of external control clients to a Halogen or HAL Control Server. This means that when you are developing and testing your control system with a commercial product, such as an AMX system, you can also connect to the control server using PuTTY.

**NOTE:** This lets you monitor most of the interaction between your external control system and your Halogen/HAL Control Server. This works because when external controls change, the Halogen/HAL Control Server sends these changes to all connected external control clients. So, when your external controller changes a control on the HAL, the HAL control server sends the change to your PuTTY client.

To use PuTTY in this way, first connect PuTTY to the Halogen/HAL Control Server that you want to monitor. Next connect your control system to the same Halogen/HAL Control Server. As your external control system interacts with the HAL system, external control messages will appear in the PuTTY window, giving you information about the operation of your external control system.

AMX® and the AMX logo are registered trademarks of AMX.
Stardraw Control is a trademark of Stardraw.com Ltd.
Crestron® is a registered trademark of Crestron Electronics, Inc.
Lexan® is a registered trademark of General Electric.
Decora® is a registered trademark of Leviton.
Windows® is a registered trademark of Microsoft Corporation in the United States and other countries.
Apple, Mac, Macintosh, iTunes, Safari, QuickTime, GarageBand, and OS X are registered trademarks of Apple Inc., registered in the U.S. and other countries.
Halogen is a trademark of Rane Corporation.