

# Polycom<sup>®</sup> SoundStructure Studio Version 1.4

This document introduces the new features of SoundStructure Studio v1.4 including the new Preset recorder and headset interface adapter support.

## 1 New Features in version 1.4

SoundStructure Studio Version 1.4.0 includes the following new features:

- Preset recorder that simplifies creating partial presets
- Integrated support of the Polycom Headset Interface Adapter
- Option to not auto-route signals in the matrix when adding channels and creating projects
- Support for command line arguments to automatically connect to a system when Studio is launched
- Automatically created events names now begin with “\_” to differentiate from user events
- Event copy function to simplify creating new events
- Wiring report includes physical logic pin connection information

In addition there were numerous minor user interface issues that were resolved.

## 2 Preset Recorder

SoundStructure Studio 1.4 includes a preset recorder that allows one to simply create partial presets by starting the recorder and then using SoundStructure Studio's user interface to change the project settings.

The recorded commands

1. Can become a new partial preset, or
2. Can be appended to the existing preset (and over-writing the existing preset), or
3. Can be appended to the settings of existing preset and saved as a new preset

Once the partial preset has been saved, the settings of the SoundStructure system will revert back to the settings before the partial preset recording started. To see the effects of the preset, execute the newly recorded partial preset from the presets page or execute directly using the `run` API command syntax.

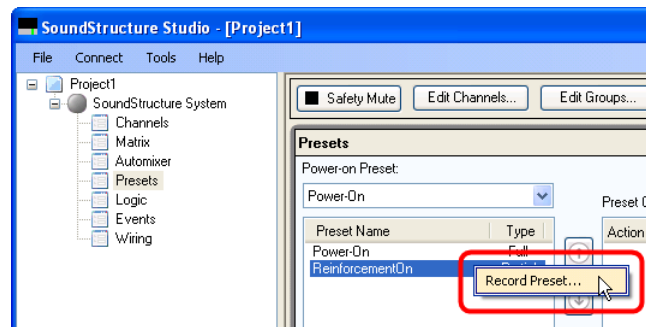
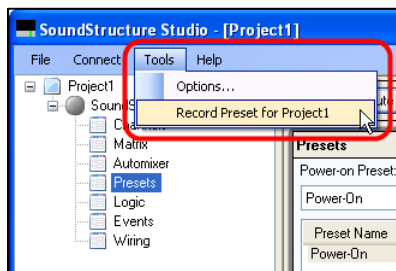
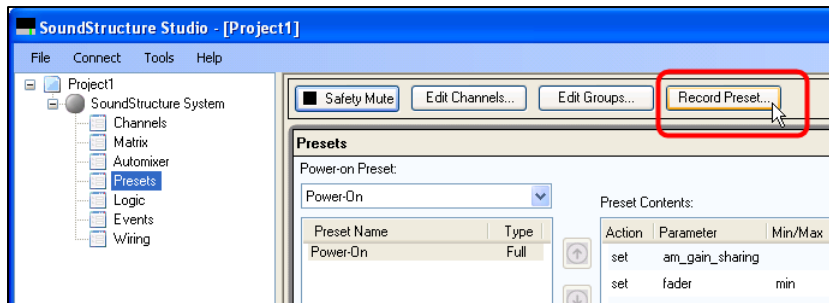


*Once a preset has been recorded, the SoundStructure Studio settings will appear as they were before the preset recording was initiated.*

To record a partial preset, follow these steps:

1. Select either <new preset> to create a new partial preset, or select an existing preset to append to
2. Click the record button
3. Make the desired changes to the project via the SoundStructure Studio user interface. Any change that can be made to the system and stored in a preset may be recorded with the preset recorder including channel page settings, matrix settings, automixer settings, etc.
4. Press the Stop button to stop recording
5. When prompted enter the name of the preset, or select the default name, and save the settings

The recorder user interface may be opened from either the Tools menu or from the presets page as shown in the following figure. If you right click on an existing partial preset, the new recorded commands will be appended to the selected preset.



**Figure 1. There are three ways to start the partial preset record – from the Record Preset button on the preset page, the Record Preset selection from the Tool menu, or by right clicking on an existing partial preset.**

Once the preset recorder option has been selected, the preset recorder user interface will appear as shown in the following figure.

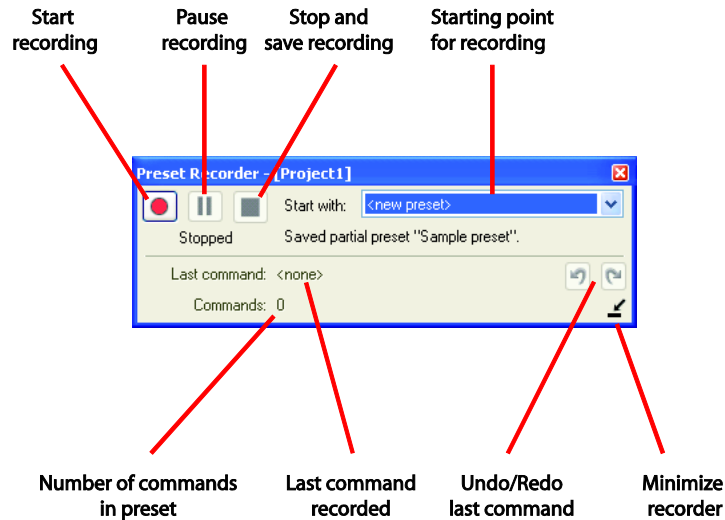


Figure 2. The user controls on the preset recorder.

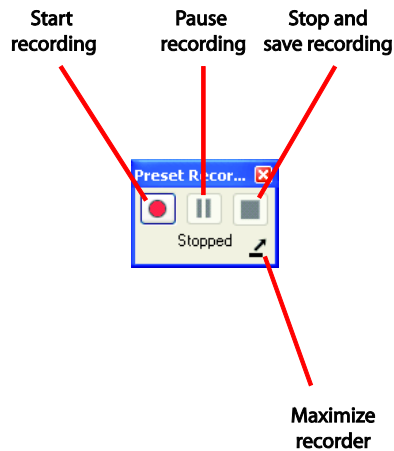
The preset recorder controls are described in the following table.

Table 1. Summary of the controls on the preset recorder window.

| Control                             | Description   |
|-------------------------------------|---|
| <i>Start recording</i>              | Initiates recording of the preset. The commands associated with subsequent Studio changes are recorded into the preset.   |
| <i>Pause recording</i>              | Pauses the recording to allow the user to change the state of the system <i>without recording the state changes into the partial preset</i> . To continue recording again press Start recording control.  |
| <i>Stop and save recording</i>      | Stops the recording. The user is prompted for the partial preset name. If recording started with a new preset, the recorded settings will be saved in the specified preset name.<br><br>If recording started with an existing preset, the recorded settings will be appended to the existing preset and the resulting preset saved in the specified name. |
| <i>Starting point for recording</i> | The preset recorder may record new partial presets or record into an existing partial preset. If an existing preset name is selected, the commands will be appended to the existing preset.   |

|                              |   |
|------------------------------|---|
| <i>Undo/Redo</i>             | Undo allows the user to remove the last command from the partial preset. Redo adds the last command back to the partial preset. |
| <i>Last command recorded</i> | Shows the last command that was recorded into the partial preset.   |
| <i>Minimize recorder</i>     | Minimizes the size of the recording window.   |
| <i>Number of commands</i>    | Keeps a running count of the number of commands in the partial preset.  |

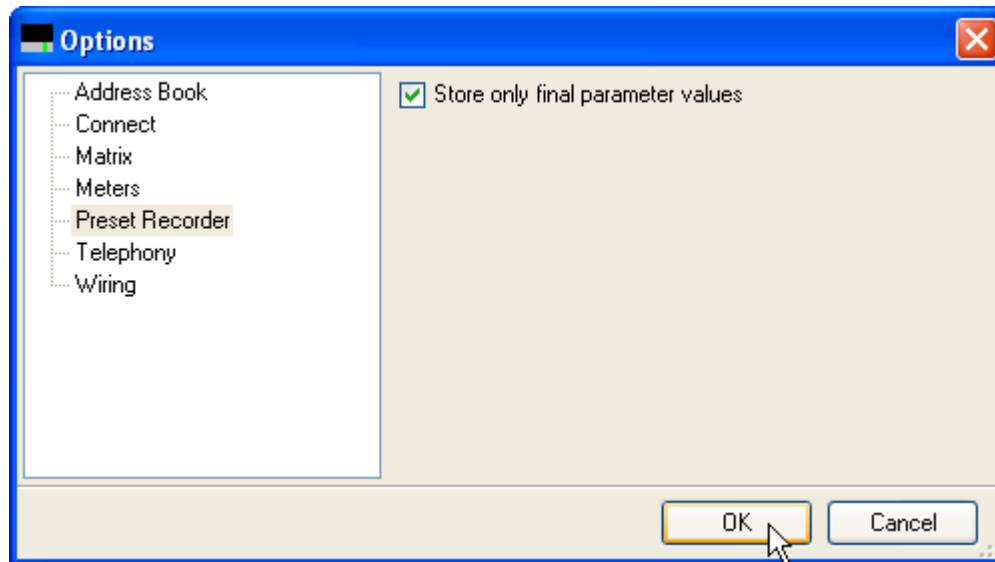
When the recorder is minimized, it will appear as shown in the following figure.



**Figure 3. The user controls on the minimized recorder.**

### Storing only final parameter values

By default, the Preset Recorder only records the final value of a given parameter. For example, if a gain slider is adjusted, only the final value of the gain parameter will be stored in the preset. If it is desired to store all the intermediate values, uncheck the Store only final parameter values on the Tools->Options->Preset Recorder menu. By default only final values are recorded in partial presets.



**Figure 4. The tools menu has a new option for only storing the final parameter. Default behavior is to only record the final parameter values.**

For example, the default behavior of the Preset Recorder would take this sequence of commands generated by moving a channel fader:

```
set fader "Table Mic 1" -1
set fader "Table Mic 1" -2
set fader "Table Mic 1" -3
set fader "Table Mic 1" -4
set fader "Table Mic 1" -5
```

and store only the last value in the preset as:

```
set fader "Table Mic 1" -5
```

If the user turned off the "Store only final parameter values" behavior, the preset would be stored as:

```
set fader "Table Mic 1" -1
set fader "Table Mic 1" -2
set fader "Table Mic 1" -3
set fader "Table Mic 1" -4
set fader "Table Mic 1" -5
```

### 3 Headset Interface Support

The Polycom Headset Interface (2200-33206-001) is directly supported within SoundStructure Studio version 1.4.

The headset interface is used to integrate a desktop phone to a SoundStructure device by connecting the phone's headset interface port to the headset adapter and connecting the analog ports on the headset adapter to an analog input and output on the SoundStructure device as shown in the following figures.

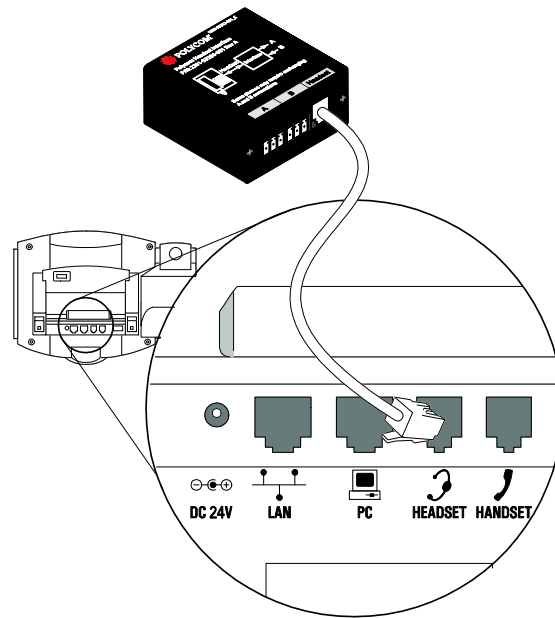


Figure 5. The headset interface connects to the RJ9 connector on the phone.

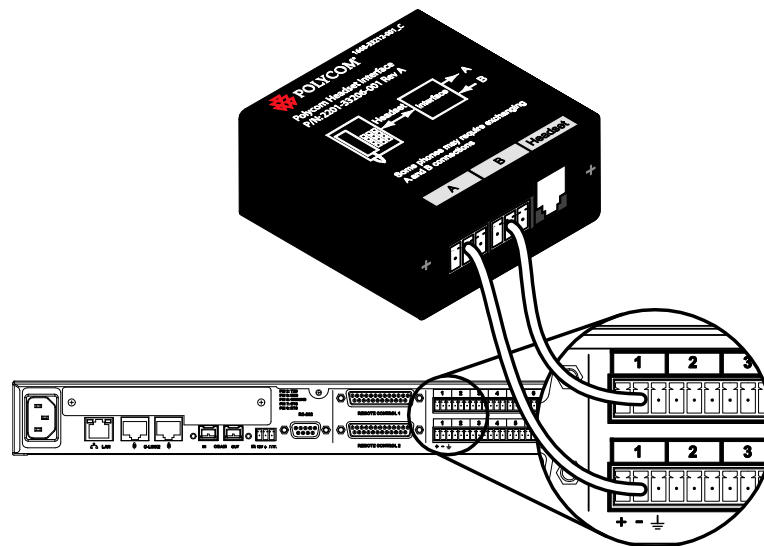


Figure 6. The headset interface connects to an analog input and output on the SoundStructure device.

To add the headset interface to a SoundStructure Studio project, select the headset interface directly from the input options presented during Step 1 of the design process and add the headset interface to the SoundStructure project.

SoundStructure Studio will automatically create the recommended settings for the headset interface including the matrix settings and the input and output channel settings.

See the Polycom Headset Interface Adapter User Manual for additional information on the settings.

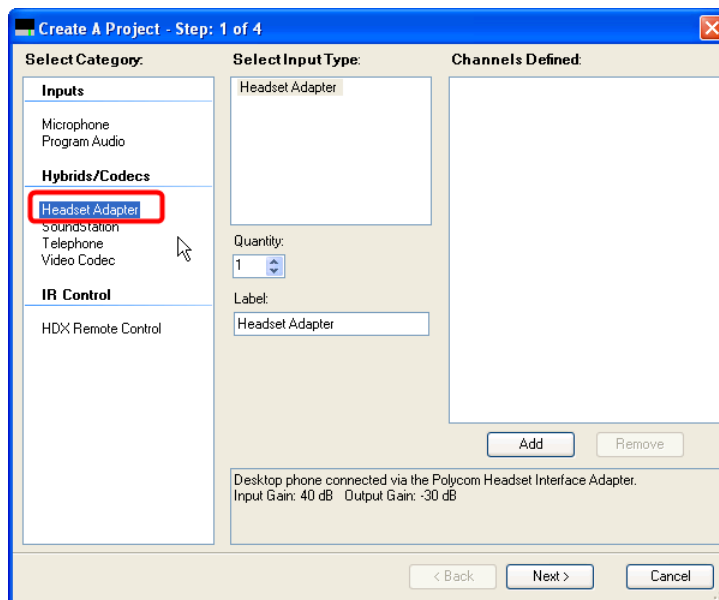


Figure 7. The Polycom headset adapter can be added to a project by selecting the headset adapter option.

## 4 Matrix Auto-routing options

SoundStructure Studio 1.4 includes a user configurable option under the Tools → Matrix settings to enable or disable automatic matrix routing of channels as they are added to the SoundStructure project.

The default option is to auto-route channel on the matrix when channels are added to a project.

When the auto-route option is *disabled*, SoundStructure Studio will not add default crosspoints to the matrix when channels are added to the design.

In either auto-route setting, SoundStructure Studio will configure the appropriate channel settings any default gains, equalization, acoustic echo cancellation settings, etc.

The Enable automatic matrix routing option retains its value between SoundStructure Studio sessions.



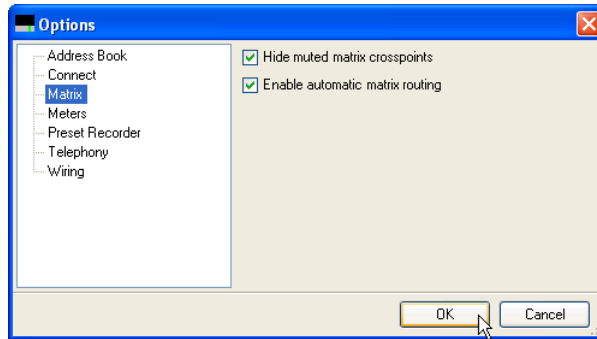


Figure 8. Option for enabling or disabling the automatic matrix routing.

## 5 Support for command line arguments

The SoundStructure Studio executable program, Studio.exe, accepts a **-connect** parameter that will automatically attempt to connect to the SoundStructure system indicated by the argument. The argument may be an IP address, a resolvable hostname, or a formatted specification of a COM port on the PC.

The IP argument is a valid IP address or hostname.

For example, the commands:

```
Studio.exe -connect 192.168.1.103
```

```
Studio.exe -connect sstr.example.com
```

will automatically launch SoundStructure Studio and attempt to connect to the specified system. If the system can be found, SoundStructure Studio will show that it is connected to the device.

The COM port argument must use the following form:

```
COMn[,b[,f]]
```

where n is a valid COM port number on the system in the range [1..256],

b is an optional baud rate in the range [9600, 19200, 38400, 57600, 115200],

and f is an optional flow control flag in the range [0, 1] with 0 indicating no flow control, and 1 indicating hardware RTS/CTS flow control.

Some examples include:

```
Studio.exe -connect COM1,19200,1           (COM1, 19200 baud rate, flow control enabled)
```

```
Studio.exe -connect COM7,57600           (COM7, 57600 baud rate, flow control off)
```

```
Studio.exe -connect COM13                 (COM13, 9600 baud rate, flow control off)
```

If no baud rate is supplied, 9600 baud is assumed. If no flow control flag is supplied, no flow control is assumed.

If the arguments are misformatted, SoundStructure Studio will start up normally and no error will be issued. If the IP address or hostname or COM port does not exist or does not respond, Studio will display the usual connection error message after trying to connect.

## 6 Automatically created events names begin with “\_”

SoundStructure Studio 1.4 creates events for HDX codec integration or HDX microphone integration with an underscore, “\_”, as the first character of the event name. The addition of the underscore makes it easier to differentiate between events generated by the system automatically and events created by the user. As with any event name, the automatically generated event names may be edited and changed to other names and the underscore may be manually removed if desired.

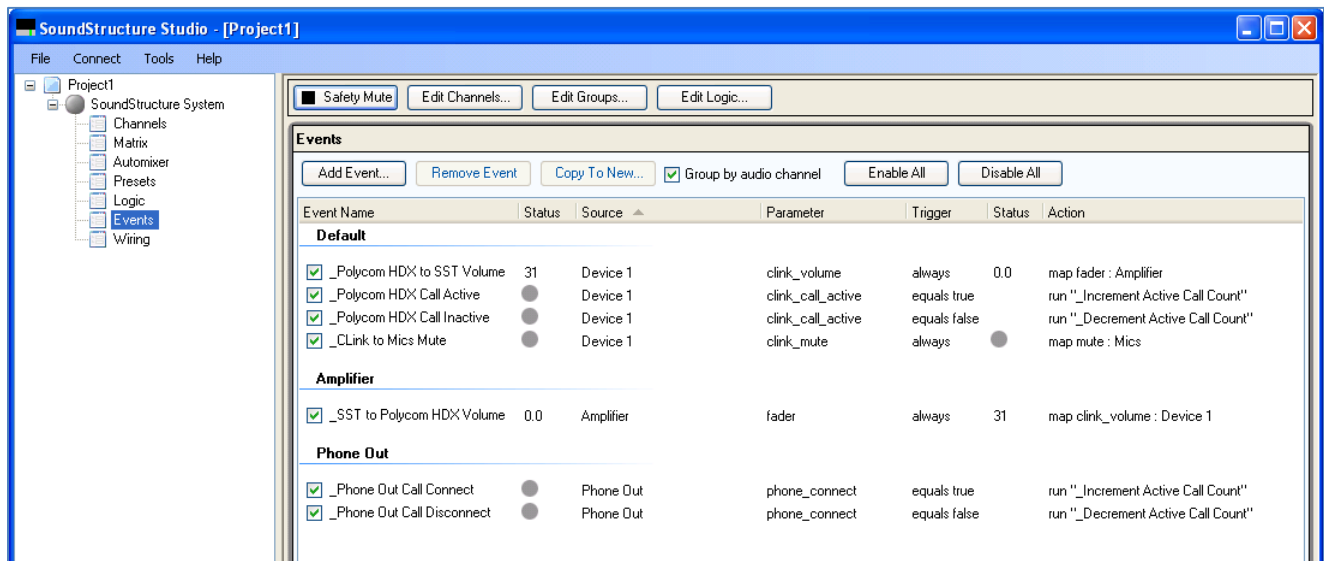


Figure 9. Events that are created by Studio now start with an “\_” character.

## 7 Event copy function simplifies creating new events

The Events page includes a Copy To New button that can be used to create a copy of an event. To copy an event, select the event by left clicking on the event name and click “Copy to New”. This will create a new event with a similar but different name from the original event and open the new event for editing.

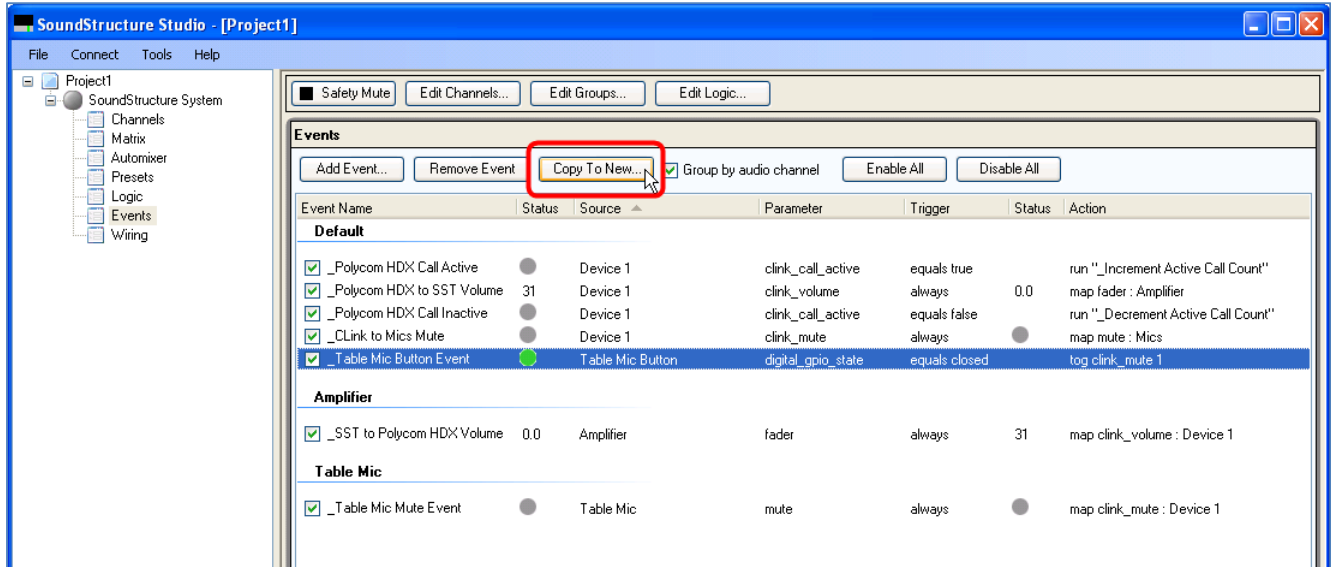


Figure 10. The Copy to New event control makes it easy to copy events.

## 8 Wiring report includes physical logic pin information

The wiring report available from the Wiring page in SoundStructure Studio includes the physical pin out of the logic pins.

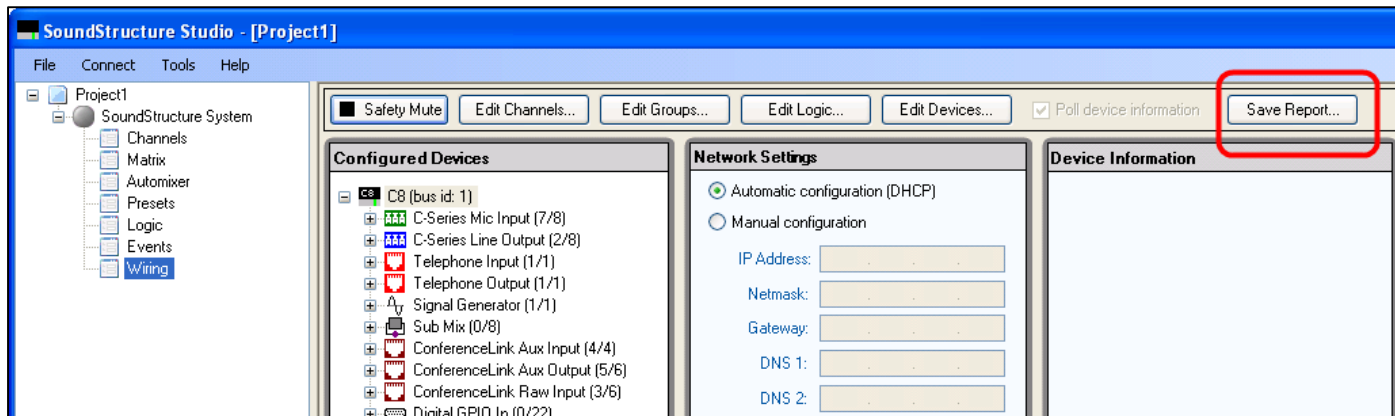
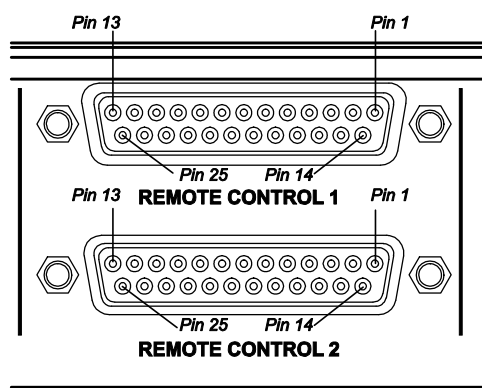


Figure 11. The Save Report button on the wiring page will create a wiring report of the system.

The Remote Control 1 and Remote Control 2 connectors on the SoundStructure rear panel contain all the logic input and output pins as shown in the following figure.



| SoundStructure Logic    |                     |     |                |
|-------------------------|---------------------|-----|----------------|
| Pin                     | Signal              | Pin | Signal         |
| <b>REMOTE CONTROL 1</b> |                     |     |                |
| 1                       | +5V                 | 14  | Logic input 1  |
| 2                       | Logic output 1      | 15  | Logic input 2  |
| 3                       | Logic output 2      | 16  | Logic input 3  |
| 4                       | Logic output 3      | 17  | Logic input 4  |
| 5                       | Logic output 4      | 18  | Logic input 5  |
| 6                       | Logic output 5      | 19  | Logic input 6  |
| 7                       | Logic output 6      | 20  | Logic input 7  |
| 8                       | Logic output 7      | 21  | Logic input 8  |
| 9                       | Logic output 8      | 22  | Logic input 9  |
| 10                      | Logic output 9      | 23  | Logic input 10 |
| 11                      | Logic output 10     | 24  | Logic input 11 |
| 12                      | Logic output 11     | 25  | Ground         |
| 13                      | Analog gain input 1 |     |                |
| <b>REMOTE CONTROL 2</b> |                     |     |                |
| 1                       | +5V                 | 14  | Logic input 12 |
| 2                       | Logic output 12     | 15  | Logic input 13 |
| 3                       | Logic output 13     | 16  | Logic input 14 |
| 4                       | Logic output 14     | 17  | Logic input 15 |
| 5                       | Logic output 15     | 18  | Logic input 16 |
| 6                       | Logic output 16     | 19  | Logic input 17 |
| 7                       | Logic output 17     | 20  | Logic input 18 |
| 8                       | Logic output 18     | 21  | Logic input 19 |
| 9                       | Logic output 19     | 22  | Logic input 20 |
| 10                      | Logic output 20     | 23  | Logic input 21 |
| 11                      | Logic output 21     | 24  | Logic input 22 |
| 12                      | Logic output 22     | 25  | Ground         |
| 13                      | Analog gain input 2 |     |                |

Figure 12. Logic pin out and association of the virtual logic pins to the physical pin out.

The new wiring report includes the physical connections such as Remote Control: 1, Pin 14 as shown in the figure below.

#### Digital GPIO In

```
1: Table Mic 1 Button (Remote Control: 1, Pin: 14)
2: Table Mic 2 Button (Remote Control: 1, Pin: 15)
3: Table Mic 3 Button (Remote Control: 1, Pin: 16)
4: Table Mic 4 Button (Remote Control: 1, Pin: 17)
5: Table Mic 5 Button (Remote Control: 1, Pin: 18)
6: Table Mic 6 Button (Remote Control: 1, Pin: 19)
7: Table Mic 7 Button (Remote Control: 1, Pin: 20)
8: Table Mic 8 Button (Remote Control: 1, Pin: 21)
9: Table Mic 9 Button (Remote Control: 1, Pin: 22)
10: Table Mic 10 Button (Remote Control: 1, Pin: 23)
11: Table Mic 11 Button (Remote Control: 1, Pin: 24)
12: Table Mic 12 Button (Remote Control: 2, Pin: 14)
13: Table Mic 13 Button (Remote Control: 2, Pin: 15)
14: Table Mic 14 Button (Remote Control: 2, Pin: 16)
15: Table Mic 15 Button (Remote Control: 2, Pin: 17)
16: Table Mic 16 Button (Remote Control: 2, Pin: 18)
```

#### Digital GPIO Out

```
1: Table Mic 1 LED (Remote Control: 1, Pin: 2)
2: Table Mic 2 LED (Remote Control: 1, Pin: 3)
3: Table Mic 3 LED (Remote Control: 1, Pin: 4)
4: Table Mic 4 LED (Remote Control: 1, Pin: 5)
5: Table Mic 5 LED (Remote Control: 1, Pin: 6)
6: Table Mic 6 LED (Remote Control: 1, Pin: 7)
7: Table Mic 7 LED (Remote Control: 1, Pin: 8)
8: Table Mic 8 LED (Remote Control: 1, Pin: 9)
9: Table Mic 9 LED (Remote Control: 1, Pin: 10)
10: Table Mic 10 LED (Remote Control: 1, Pin: 11)
11: Table Mic 11 LED (Remote Control: 1, Pin: 12)
12: Table Mic 12 LED (Remote Control: 2, Pin: 2)
13: Table Mic 13 LED (Remote Control: 2, Pin: 3)
14: Table Mic 14 LED (Remote Control: 2, Pin: 4)
15: Table Mic 15 LED (Remote Control: 2, Pin: 5)
16: Table Mic 16 LED (Remote Control: 2, Pin: 6)
```

Figure 13. Example of the logic pin out in the wiring report.