

# Escape Room Software

## User guide

Escape Room Software - User guide

All rights reserved.

© Escape Room Software 2016-2025

Windows is a registered trademark of Microsoft Corporation in the United States and other countries.



## Table of contents

1. Introduction.....	6
2. Feature overview.....	7
2.1. Basic.....	7
2.2. Customization.....	7
2.3. Client interaction.....	7
2.4. Technical.....	7
3. Setting up.....	8
3.1. Single computer setup.....	8
3.2. Multiple computer setup.....	9
3.3. Hardware requirements.....	10
3.4. Recommended update strategy.....	10
4. Getting started.....	11
4.1. Step 1.....	11
4.2. Step 2.....	11
4.3. Step 3.....	11
4.4. Step 4.....	12
4.5. Step 5.....	12
5. Screens overview.....	13
5.1. Start screen.....	13
5.2. Settings.....	14
5.2.1. General.....	14
5.2.2. Advanced.....	15
5.2.3. License.....	16
5.3. Statistics.....	17
5.3.1. General statistics.....	17
5.3.2. Sessions.....	17
5.3.3. Action counters.....	18
5.4. Multi-room mode.....	19
5.5. Room settings screen.....	20
5.5.1. General.....	20
5.5.2. Export room.....	21
5.5.3. Connections.....	22
5.5.4. Self-Hosted API.....	23
5.5.5. Home Automation Integration with Z-Wave.....	24
5.5.6. External controllers.....	24

5.5.7.	Server screen.....	25
5.5.8.	Client screen.....	26
5.5.9.	Languages.....	27
5.6.	Game master panel.....	28
5.6.1.	Logfile window .....	29
5.7.	New room wizard dialog .....	30
5.7.1.	General.....	30
5.7.2.	Modify room layout.....	31
5.7.3.	Atmosphere.....	32
5.7.4.	Actions.....	33
5.8.	Action dialog .....	34
5.8.1.	Cause .....	34
5.8.2.	Timed, based on PC time schedule .....	35
5.8.3.	Keypad entry .....	35
5.8.4.	Effect .....	36
5.8.5.	Advanced.....	38
5.9.	Advanced script editor.....	39
5.10.	In-room client screen .....	41
5.11.	Limiting access on the client computer .....	41
5.12.	Importing a room on a dedicated client computer .....	42
6.	Scripting guide .....	43
6.1.	General.....	43
6.2.	Script types.....	43
6.3.	General script definitions.....	43
6.4.	Examples .....	44
7.	Function glossary.....	45
7.1.	MessageActions .....	45
7.2.	TimerActions .....	46
7.3.	ImageActions.....	47
7.4.	AudioActions .....	48
7.5.	VideoActions .....	49
7.6.	BackgroundLayout .....	50
7.7.	TextLayout.....	51
7.8.	ClientInput .....	55
7.9.	Scripting .....	56
7.10.	Connectivity .....	57

7.11.	HomeAutomation .....	58
7.12.	Variables.....	59
7.13.	Reserved.....	60
7.14.	System variables .....	60
7.15.	Incoming messages from external devices .....	60
8.	NET colors overview .....	61
8.1.	Color chart.....	61
9.	How to connect to a micro controller .....	62
9.1.	Arduino .....	62
9.1.1.	API connection .....	62
9.1.1.1.	Send API call from Arduino to ERS Server .....	62
9.1.1.2.	Send API call from ERS Server to Arduino .....	63
9.1.2.	TCP ethernet connection (not recommended, please use API).....	64
9.1.3.	TCP serial connection (not recommended, please use API) .....	66
9.2.	Raspberry PI .....	67
9.2.1.	Send API call from Raspberry PI to ERS Server.....	67
9.2.2.	Send API call from ERS Server to Raspberry PI.....	67
9.2.3.	Automatically start Raspberry PI software upon startup.....	68
10.	Frequently Asked Questions (FAQ) .....	69
10.1.	How can I release my license key for migration to a new computer?.....	69
10.2.	Is there an Intercom/Camera feature.....	69
10.3.	Suddenly my client shows a black screen, doesn't respond or behaves in a strange fashion 69	
10.4.	I want to have a 3 <sup>rd</sup> computer+screen in my room that is connected to the Game Master PC (Server), but behaves differently .....	69





## 1. Introduction

Escape Room Software is a software solution for real escape rooms. Basically it displays a countdown timer on one or multiple screens in an escape room so your players know how much time is left in the game.



### **Do more with ambiance**

But you can do so much more! Perhaps influence the mood and ambiance by playing music. You can send text messages when players need a hint. Or weave an intricate storyline by sending videos or images to the screen at appropriate times.



### **Do more with player interaction**

Add to that some advanced interaction possibilities: if the in-room screen is a touchscreen, you can provide players with a hint button or perhaps a quiz to earn extra time at a specific junction in the game.



### **Do more with external interaction**

Escape Room Software works with a simple script editor that you can use to create your own preset sequences. These scripts can also be kicked off by external input, for example by the players pressing a button on the touchscreen, or an advanced room prop connected to a microcontroller-device (for example an Arduino), specific Home Automation Solutions, or even external websites (using an API).












### **Focus on what makes your room special!**

Escape Room Software avoids complex technical dependencies for your core game. Suitable for single rooms as well as managing multiple rooms from a central chamber.









In this way, Escape Room Software can become the solid backbone of your escape game and provide a high level of user interactivity, increasing the immersiveness for both the player and the game master.

## 2. Feature overview








### 2.1. Basic

Description	Details
 Countdown timer	Hours, minutes or cyclic
 Play intro movie	Optional, autostart timer after finishing movie
 Send messages	Manually using buttons, or scripted, colored
 Send alerts	Manually using buttons, or scripted
 Send images	Supported filetypes (.jpg, .jpeg, .png, .bmp)
 Play audiofiles	Supported filetypes (.mp3, .wav)
 Play & skip through video files	Supported filetypes (.avi, .mp4, .m4v, .mov), timer sync.
 Manipulate countdown timer	Manually using buttons, or scripted, movie sync.
 Scary flicker effect	





### 2.2. Customization

Description	Details
 Customizable layout	Texts, images, fonts, colors, sizes, locations
 Easy-to-use wizard & scripting tools	
 Manual action scripts	
 Timed action scripts	
 Variables	Define, update, compare, randomize
 Save customized rooms as .ZIP file	
 Scripting functions glossary	Also available in-software
 Sample room included	Showing various functions

### 2.3. Client interaction

Description	Details
 Client-side touchscreen buttons	
 Client-side keyboard/pad/barcode scanner support	Requires keyboard, numpad or (USB)barcode scanner
 Run external processes on client or server	
 Accepts external Serial or TCP commands to kick off scripts	<i>For example:</i> messages sent by an external application, or (micro)controllers like Arduino or Raspberry Pi
 Send a custom Serial or TCP command	<i>For example:</i> to send messages to an external application, or (micro)controllers like Arduino or Raspberry Pi
 Home Automation integration	Z-Wave integration
 API	Receive and send API calls

### 2.4. Technical

Description	Details
 Simple MSI installer	
 Host one or multiple rooms using one or more devices	One computer for SERVER and CLIENT or use one server computer and multiple clients
 Simple TCP broadcasting	TCP server autodiscovery is supported
 For Windows 7, 8 and 10 (.NET 4.5 required)	Please note that WinRT is not supported

### 3. Setting up

#### 3.1. Single computer setup

If you have just one room you can set up using just one Windows computer.

Single computer setup	Details
<b>Server</b>	Use one computer as both server and client
<b>Client (screens)</b>	<p>Use a second computerscreen, place it in a room and, connect with a long HDMI cable.</p> <p>Use a MiraCast solution which can replicate a Windows desktop on the screen.</p> <p>For example: MiraCast is just a wireless HDMI cable. This makes a regular TV monitor behave as an actual second screen of your server computer.</p>
<b>Drawbacks</b>	<p>No or limited touchscreen possibilities.</p> <p>Multiple rooms on different screens of one computer is tricky. You could get it to work using one computer, but you'll probably end up with multiple server-side computers due to audio restraints.</p>

Common examples of single computer setups:

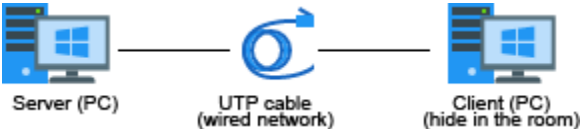



Description	Requirements	Example
<b>PC, classic</b>	<ul style="list-style-type: none"> <li>1 PC</li> <li>Second screen</li> <li>1 HDMI cable</li> </ul>	<p>The diagram shows a desktop computer labeled 'Server (PC)' connected to a monitor labeled 'Second screen' by a cable labeled 'HDMI cable'.</p>
<b>PC, wireless MiraCast</b>	<ul style="list-style-type: none"> <li>1 PC (MiraCast enabled)</li> <li>Second screen</li> <li>1 MiraCast adapter</li> </ul>	<p>The diagram shows a desktop computer labeled 'Server (PC) (MiraCast enabled)' connected to a monitor labeled 'Second screen' via a 'Wireless MiraCast' connection (represented by a Wi-Fi symbol) and a 'MiraCast adapter' connected to the monitor.</p>
<b>Tablet, wireless MiraCast</b>	<ul style="list-style-type: none"> <li>1 tablet (MiraCast enabled)</li> <li>Second screen</li> <li>1 MiraCast adapter</li> </ul>	<p>The diagram shows a tablet labeled 'Server (tablet) (MiraCast enabled)' connected to a monitor labeled 'Second screen' via a 'Wireless MiraCast' connection (represented by a Wi-Fi symbol) and a 'MiraCast adapter' connected to the monitor.</p>

### 3.2. Multiple computer setup

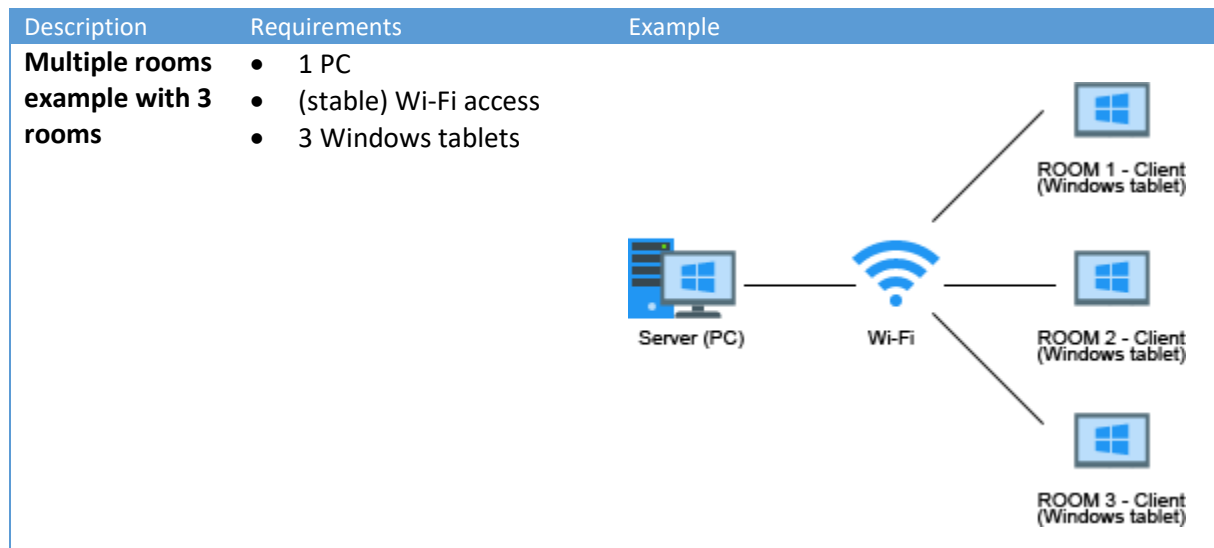
If you have multiple rooms, please consider using multiple Windows computers or Windows tablets.

Multiple computer setup	Details
<b>Server</b>	Use one server computer to host multiple room clients.
<b>Client (screens)</b>	<p>Use actual computers in the rooms as clients. Usually you want tablets for this because they are small, easy to attach, sleek and have touchscreens. They must be tablets with Windows 10.</p> <p>An alternative is to hide a computer in the room where it is not accessible to the players and place the screen somewhere near it.</p> <p>Please note that a network connection (cabled or wifi) is required.</p>
<b>Drawbacks</b>	Multiple computers needed.

Common examples of multiple computer setups:

Description	Requirements	Example
<b>2 PC's, classic</b>	<ul style="list-style-type: none"> <li>• 2 PC's</li> <li>• Cabled network access</li> </ul>	 <p>Server (PC) — UTP cable (wired network) — Client (PC) (hide in the room)</p>
<b>2 PC's, wireless</b>	<ul style="list-style-type: none"> <li>• 2 PC's</li> <li>• (stable) Wi-Fi access</li> </ul>	 <p>Server (PC) — Wi-Fi — Client (PC) (hide in the room)</p>
<b>1 PC, 1 tablet, wireless</b>	<ul style="list-style-type: none"> <li>• 1 PC</li> <li>• 1 Windows tablet</li> <li>• (stable) Wi-Fi access</li> </ul>	 <p>Server (PC) — Wi-Fi — Client (Windows tablet)</p>
<b>2 tablets, wireless</b>	<ul style="list-style-type: none"> <li>• 2 Windows tablets</li> <li>• (stable) Wi-Fi access</li> </ul>	 <p>Server (Windows tablet) — Wi-Fi — Client (Windows tablet)</p>

Example of a setup with multiple rooms:



### 3.3. Hardware requirements

Computer requirements (for any computer used in your solution)

- Windows tablet/PC
- Windows 10 or higher is recommended
- .NET 4.8 must be installed
- Stable network access (cabled or WiFi)
- Firewall access to the configured TCP or UDP ports (you should allow access when the Windows firewall access screen pops up)
- Touchscreen is optional
- Speakers are optional
- MiraCast adapter is optional

### 3.4. Recommended update strategy

The key words are "*predictable outcome*". Set up a test environment. A computer where you have the current version installed, as well as your room.

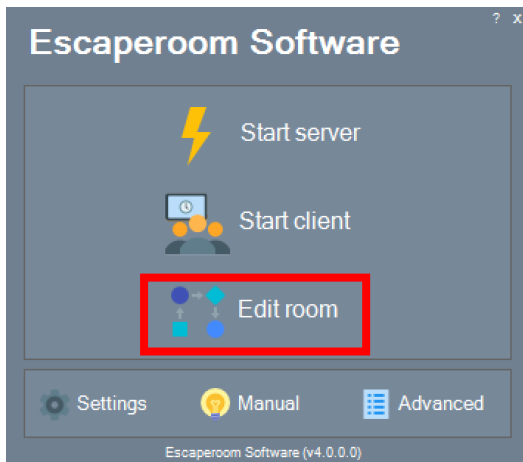
When a new version is released, download and install the update in the test environment.

In this way you will be able to check out the changes without affecting and interfering with your production (live) environment.

## 4. Getting started

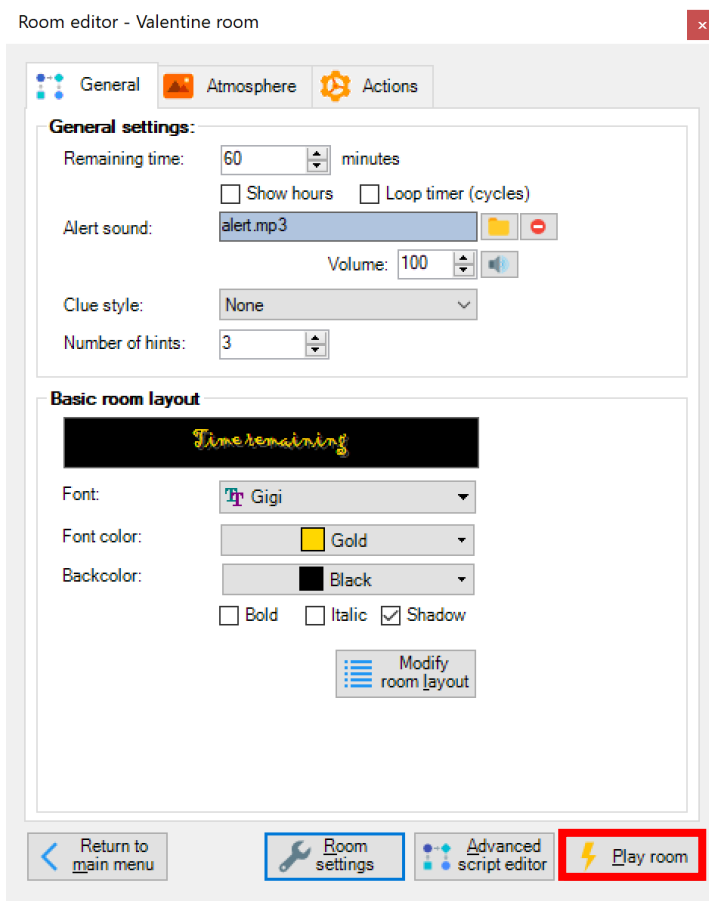
### 4.1. Step 1

Run the main menu and create a new room by pressing the EDIT ROOM button.



### 4.2. Step 2

Use the Wizard to easily configure a new and simple room.

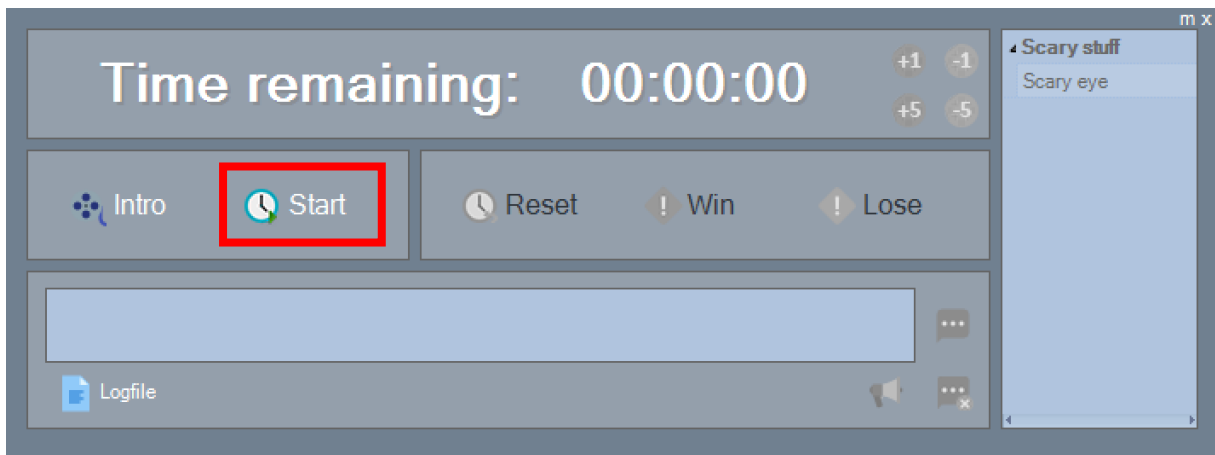


### 4.3. Step 3

Press the Play room button to test your room.

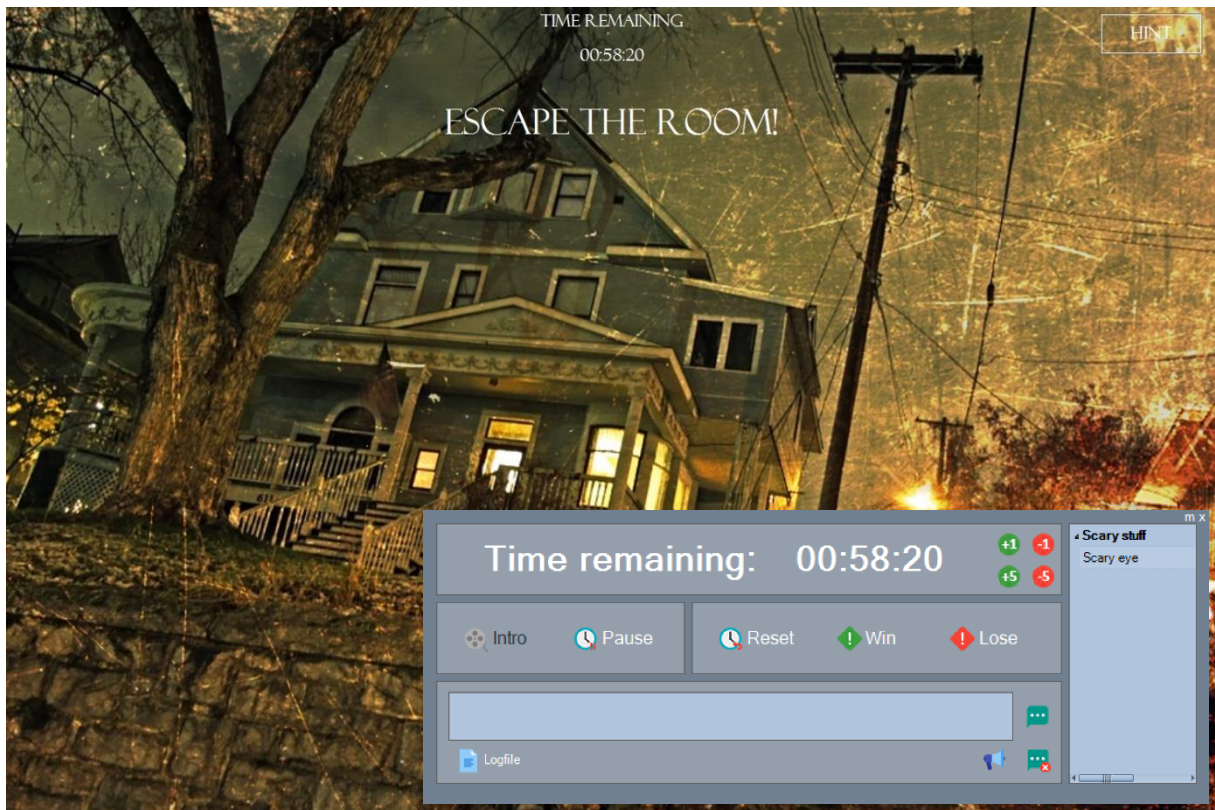
#### 4.4. Step 4

Start your newly created room using the START button.



#### 4.5. Step 5

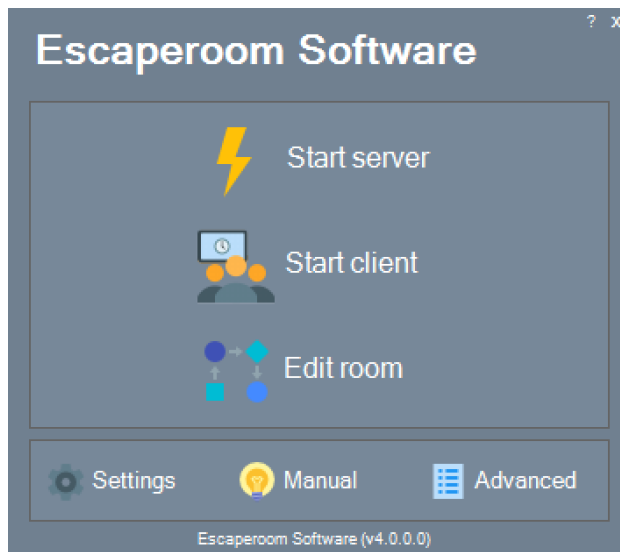
Enjoy your newly created room!



Please check the rest of this manual for more instructions and possibilities!

## 5. Screens overview

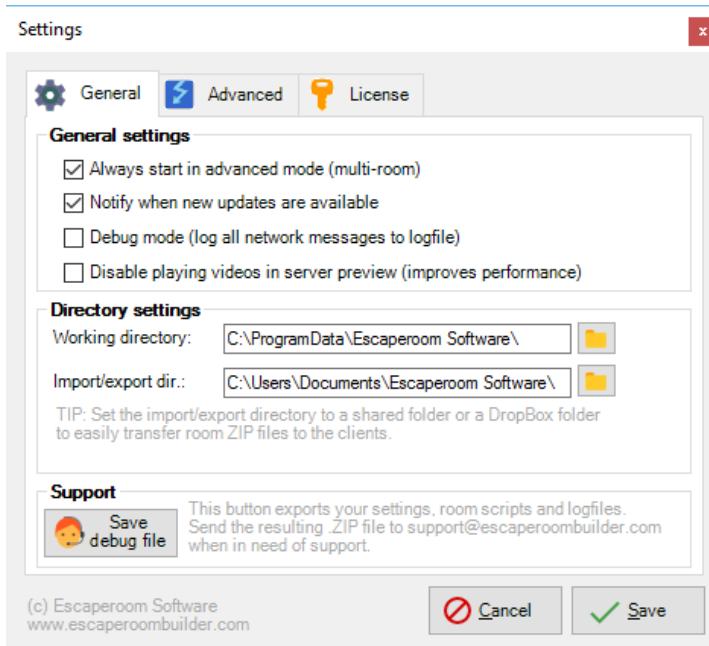
### 5.1. Start screen



Start screen	Details
Start server	Start the selected room in SERVER MODE. Please note that audio for SERVER MODE is muted per default.
Start client	Start the selected room in CLIENT MODE. Use F12 to close CLIENT MODE.
Edit room	Start the selected room in EDITOR MODE. Please note that EDIT MODE runs the room in it's configured resolution.
Settings	Go to the settings screen.
Manual	Shows the PDF help file. You can also press the F1 key.
Advanced	Open the advanced screen when using multiple rooms. You can change the settings to automatically start in multi-room mode.
Version & license info	Shows version number and licenses in use. Click to visit our website to purchase licenses or check for updates.

## 5.2. Settings

### 5.2.1. General



General settings	Details
Always start in advanced mode (multi-room)	When handling multiple rooms from one computer, you probably want to always start in multi-room mode.
Debug mode (log all network messages to logfile)	Turn this on for more detailed logging and insight in what messages are being sent and in what order. Press F10 in editor, server or client modes.
Disable playing videos in server preview (improves performance)	When running both server and client on 1 PC, videos are usually played on both client and server. This can cause performance issues if the video quality is high. This feature skips playing of videos on the server side.

Directory settings	Details
Working directory	This is the folder where active room files, scripts and resources are unzipped. The <b>programdata</b> folder is configured per default.
Import/export directory	The folder where you collect your room ZIP files. You could choose to make this a shared folder or a dropbox folder to easily transfer room ZIP files to your clients.

Support	Details
Save debug files	This button exports your settings, room scripts and logfiles. Send the resulting .ZIP file to <a href="mailto:support@escaperoombuilder.com">support@escaperoombuilder.com</a> . This button also turns on 'debug mode'.

## 5.2.2. Advanced

Settings

General Advanced License

**Intercom settings**

Input device: Microphone Array (Realtek High) ▾

Output device: Speakers (Realtek High Definiti) ▾

Please note that for intercom features, a multiple device setup is required.

**E-mail settings**

SMTP host:

SMTP port: 25 ▾

Username:

Password:

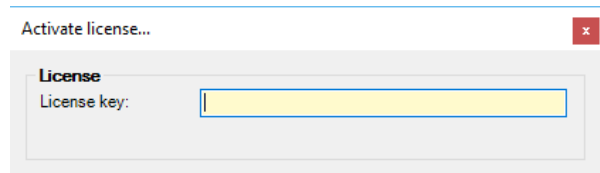
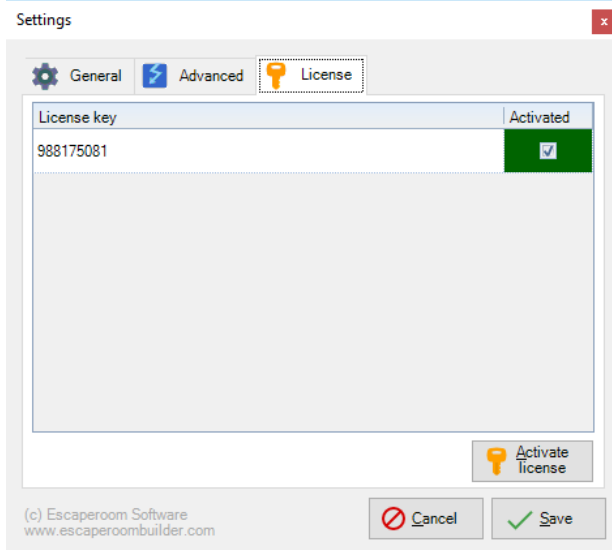
From address:

(c) Escaperoom Software  
www.escaperoombuilder.com

Cancel Save

E-mail settings	Details
SMTP host, port	Optionally configure an SMTP server. These settings enable the e-mail functionality of the scripting engine.
Username, password	Enter your credentials if your SMTP account requires authentication.
From address	Used as from and reply-to address in any e-mails sent.

### 5.2.3. License



License	Details
License key	Enter your license key here.
Activate	<p>Activate your software on-line.</p> <p>Please note that <b>you only have to activate your SERVER installation.</b></p> <p>Please note that <b>your CLIENT installation does NOT need any activation.</b></p> <p>Please note that <b>internet connection is only required during initial activation.</b></p>

## 5.3. Statistics

### 5.3.1. General statistics

Statistic description	Total	Prev. mo.	This mon.	Difference
Games played	56	0	5	100.0%
Games won	4	0	2	100.0%
Games lost	2	0	0	0.0%
Games not finished	44	0	1	100.0%
Average playing time (min)	1,142857	0	0,6	100.0%
Total playing time (min)	64	0	3	100.0%

Statistics	Details
Statistic description	Describes the statistic shown in that line.
Total	Shows the grand total of KPI results.
Previous month	Shows the KPI results for the previous month.
This month	Shows the KPI results for this month.
Difference	A percentage, comparing this month with the previous month.
Clear statistics	Reset the statistics table

### 5.3.2. Sessions

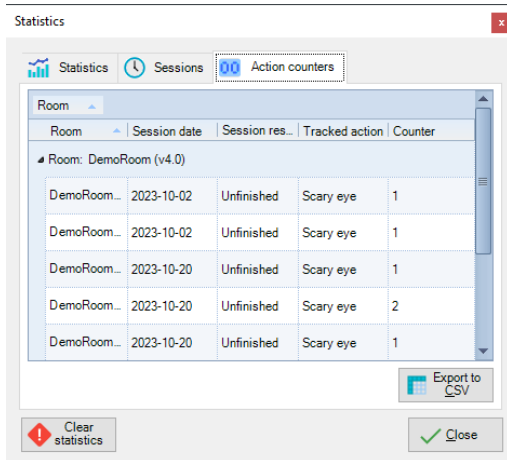
Room	Last modified	Started	Stopped	Minutes elapsed
Room: DemoRoom (v4.0)				
DemoRoom (v4.0)	2-10-2023 1..	2-10-2023 1..	2-10-2023 1..	0
DemoRoom (v4.0)	2-10-2023 2..	2-10-2023 2..	2-10-2023 2..	0
DemoRoom (v4.0)	2-10-2023 2..	2-10-2023 2..	2-10-2023 2..	0
DemoRoom (v4.0)	2-10-2023 2..	2-10-2023 2..	2-10-2023 2..	0
DemoRoom (v4.0)	2-10-2023 2..	2-10-2023 2..	2-10-2023 2..	0

Sessions	Details
Room	Name of the room.
Last modified	When the room was last modified.
Started	Game session start timestamp.
Stopped	Game session stop timestamp.
Minuteds elapsed	Game time in minutes
Game result	Win, Lose, InstaLose (=triggered loss), Unfinished
Custom 1-9	Using scripts, you can store variable values into custom fields 1-9 (see <code>SaveVariableToSessionStats</code> )
Export to CSV	Export to Comma Separated Values (CSV) file; this type of file can be imported into a Spreadsheet tool.

### 5.3.3. Action counters

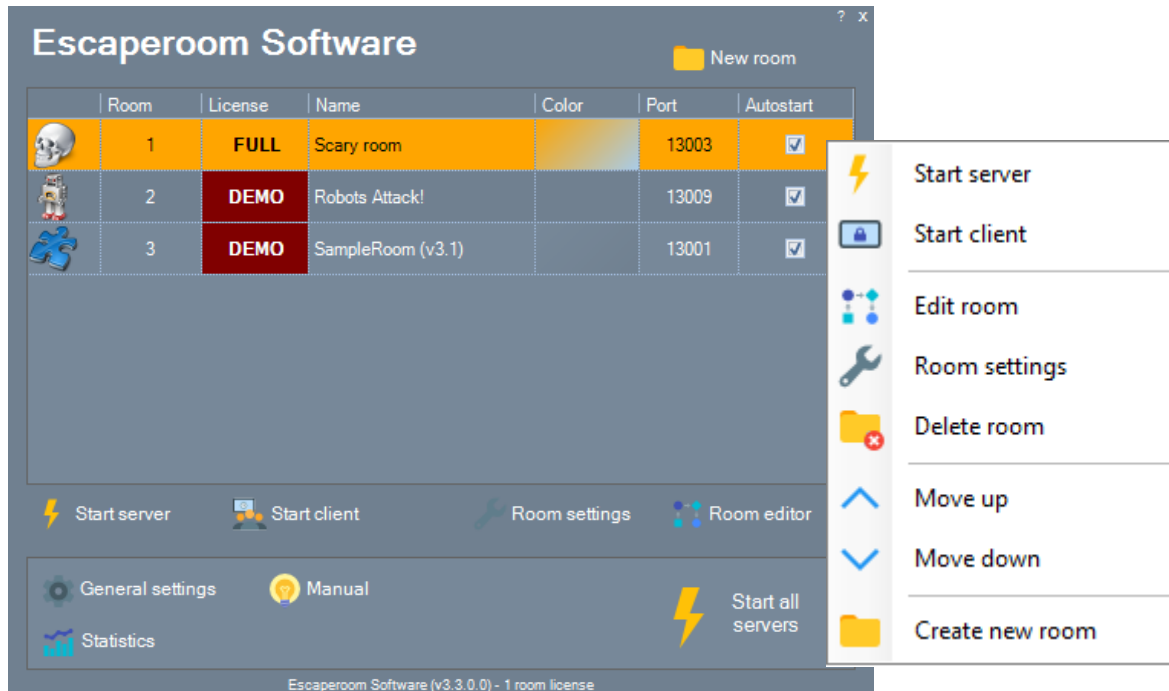
This screen shows how often certain actions or clues were used during the game sessions.

Please note that you must **turn on *Track Statistics*** in an Action to track it in this overview.



Counters	Details
Room	Name of the room.
Date	Game session date.
Result	Game session result.
Counter	The action being counted.
Value	The num
Export to CSV	Export to Comma Separated Values (CSV) file; this type of file can be imported into a Spreadsheet tool.

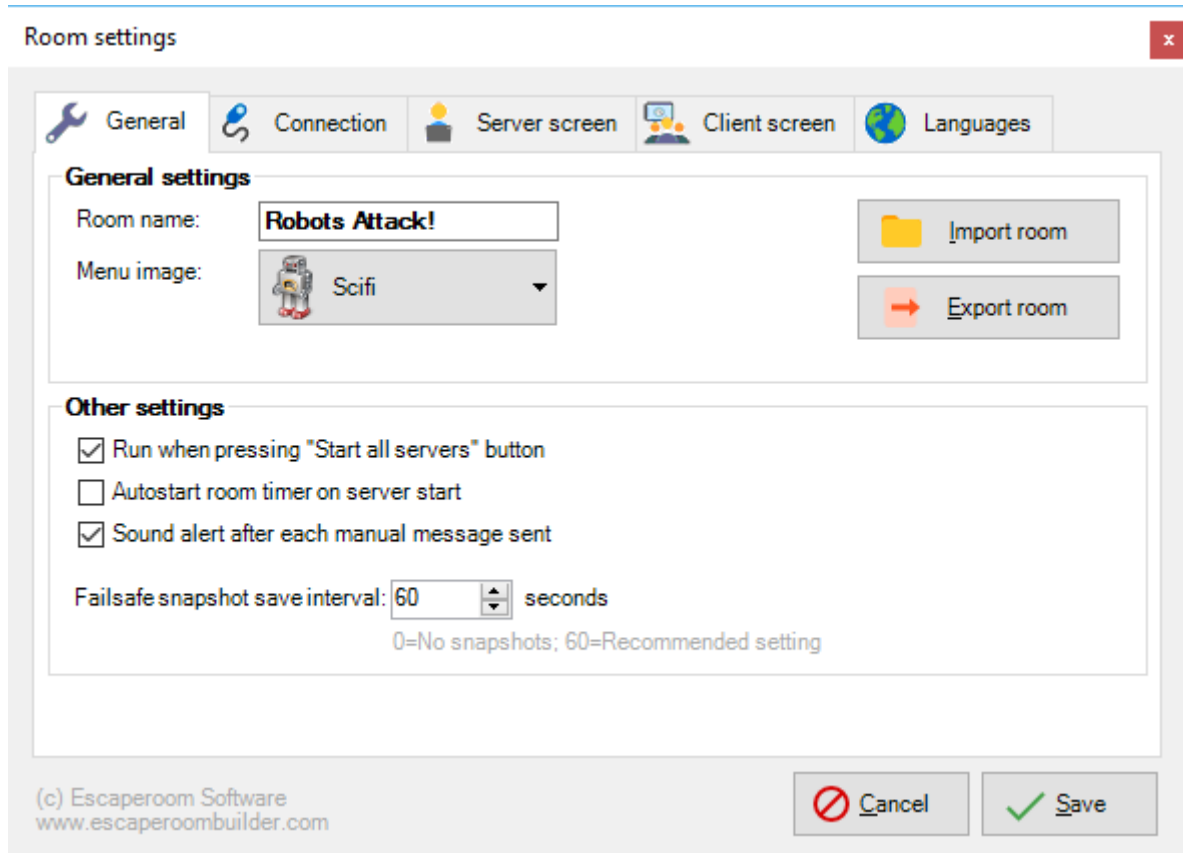
## 5.4. Multi-room mode



Main menu	Details
New room	Create a new room.
Selection grid	The selection grid shows all your rooms. Doubleclick to open the ROOM SETTINGS.
Image	A nice image to indicate the type of room.
Room	Shows the room slot number. You have licenses for a limited number of slots.
License	Shows whether a room is LICENSED or in DEMO mode.
Name	The room name.
Color	Shows the configured game master panel color for this room.
Port	Shows the configured port used for this room.
Autostart	Indicates that the room will be automatically started when pressing the START ALL SERVERS button.
Start server	Start the selected room in SERVER MODE. Please note that audio for SERVER MODE is muted per default.
Start client	Start the selected room in CLIENT MODE. Use F12 to close CLIENT MODE.
Room settings	Display the room settings dialog.
Edit room	Start the selected room in EDITOR MODE. Please note that EDIT MODE runs the room in it's configured resolution.
Move up/down	Move a room up or down a slot.
General settings	Go to the general settings screen.
Statistics	Go to the statistics screen.
Manual	Shows the PDF help file. You can also press the F1 key.
Start all servers	Starts all rooms that have "Autostart" checked.
Version & license info	Shows version number and licenses in use. Click to visit our website to purchase licenses or check for updates.

## 5.5. Room settings screen

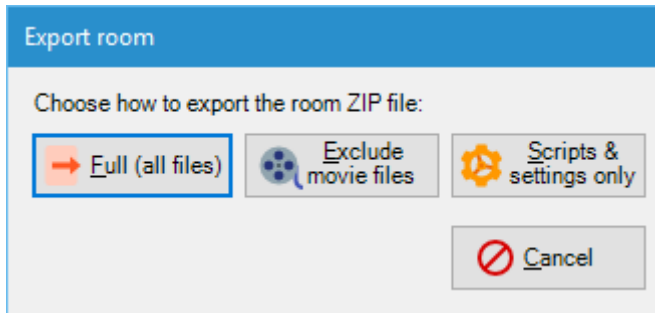
### 5.5.1. General



General settings	Details
Room name	The room name.
Menu image	A nice image to indicate the type of room.
Import room	Import a room .ZIP file for the current selected slot.
Export room	Export a room .ZIP file based on the current selected slot.

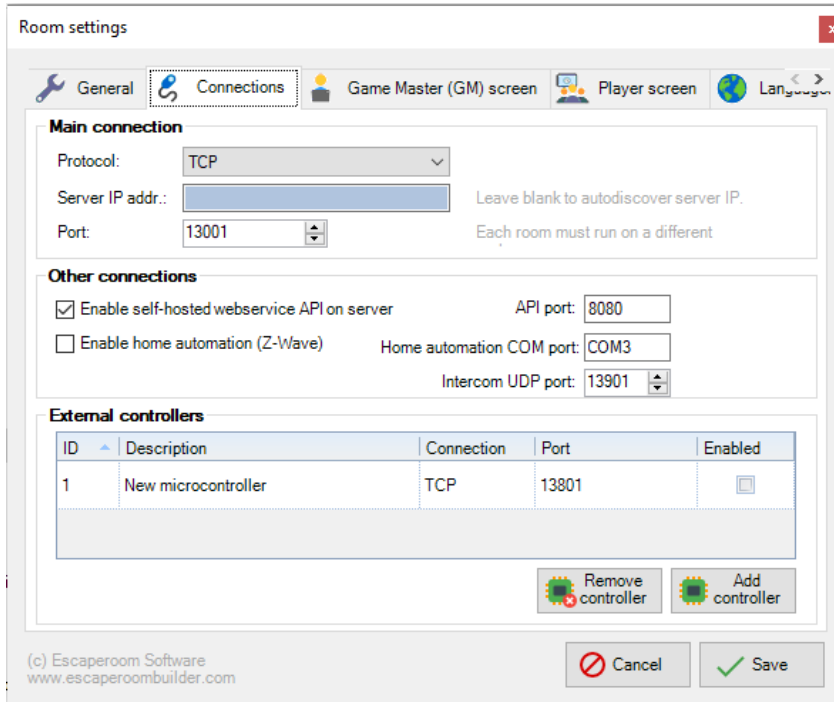
Other settings	Details
Run when pressing "Start all servers" button	Indicates that the room will be automatically started when pressing the START ALL SERVERS button.
Autostart room timer on server start	Automatically start room (timer) when starting this room in server-mode.
Sound alert after each manual message sent	Whether or not an alert sound/flash should be sent after the game master sends a message to the room.
Snapshot save interval n seconds	Save a snapshot for failsafe recovery every n seconds.  If a game session is ended improperly, on restart of that room you will be asked whether to continue based on the last saved snapshot.  0 = No snapshots 60 = Recommended setting

### 5.5.2. Export room



Export room button	Details
Full (all files)	Export a room ZIP file including all files.
Exclude movie files	Export a room ZIP file excluding all video files (possibly skipping gigabytes of movie files)
Scripts & settings only	Export a bare room ZIP file, only containing scripts and room settings. (smallest)

### 5.5.3. Connections



Main connection	Details
Connection (TCP)	Indicates the protocol that is used for connection.  Please note that UDP connection is experimental and is no longer supported for all functions.
Server IP address	Leave the server IP field blank for clients to automatically detect the server IP. For firewall purposes, please note that this feature uses a TCP/UDP combination to detect the server IP.
Port	Please note that you must configure a different port <i>for each different escape room</i> in order for the right message to reach the right room clients.

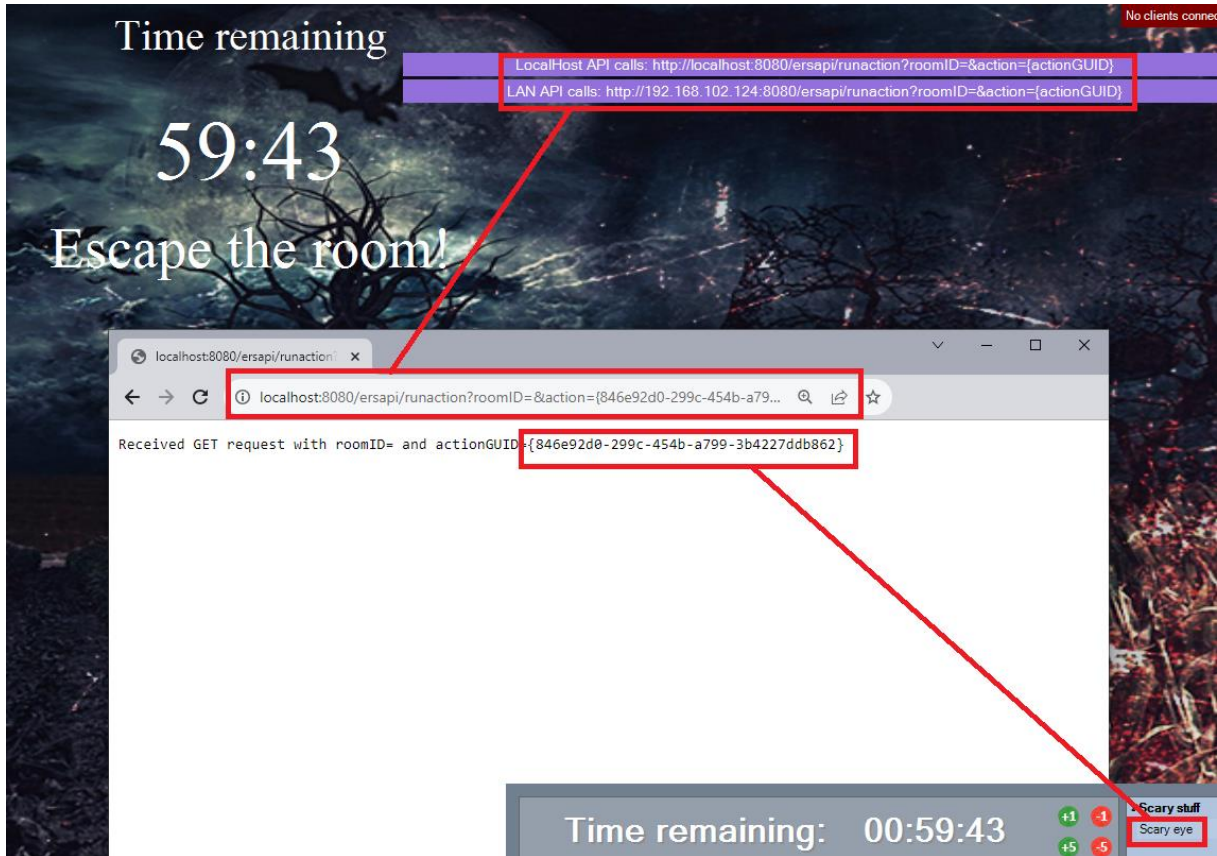
Other connections	Details
Enable self-hosted webservice API on server (you must 'run as Administrator')	Turn on self-hosted API on a port. You must run the Server 'as Administrator' to enable access.
Enable home automation integration (Z-Wave)	Turn on and configure a Z-Wave COM port to enable home automation functions.

External controllers	Details
Add controller	Add a new external controller. Allows you to configure interaction with external controllers (Arduino, Raspberry PI).
Remove controller	Remove an existing external controller.

#### 5.5.4. Self-Hosted API

Turn on self-hosted API on a port. You must run the Server 'as Administrator' to enable access.

The Server will show how to access local and LAN. You can leave roomID empty for 1 room or when notifying all rooms. The actionGUID can be found on the action dialog.



#### **Local machine API call example**

<http://localhost:8080/ersapi/runaction?roomID=&action={846e92d0-299c-454b-a799-3b4227ddb862}>

#### **LAN API call example**

<http://192.168.100.2:8080/ersapi/runaction?roomID=&action={846e92d0-299c-454b-a799-3b4227ddb862}>

Please check **chapter 9** for examples on how to connect with **microcontrollers**.

### 5.5.5. Home Automation Integration with Z-Wave

Using a Z-Wave Stick in a server USB-port you can control paired Z-Wave components.

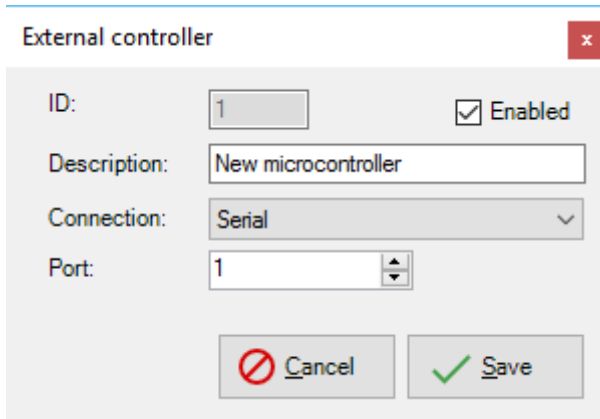
#### Supported components

- Smart plugs
- Door/window sensors

#### Troubleshooting

- Make sure the Z-Wave Stick is in a *powered* USB port
- Make sure the Z-Wave Stick COM port number is correct (this may change when switching USB ports)
- If Z-Wave devices won't connect to the Z-Wave Stick, put the Z-Wave Stick in "un-pair"-mode and keep it near the troublesome device. This should "un-pair" the device, after which you should be able to normally pair it to the Z-Wave Stick again
- Make sure the proper Z-Wave device id is chosen from the dropdown in the Escaperoom Software actions dialog

### 5.5.6. External controllers

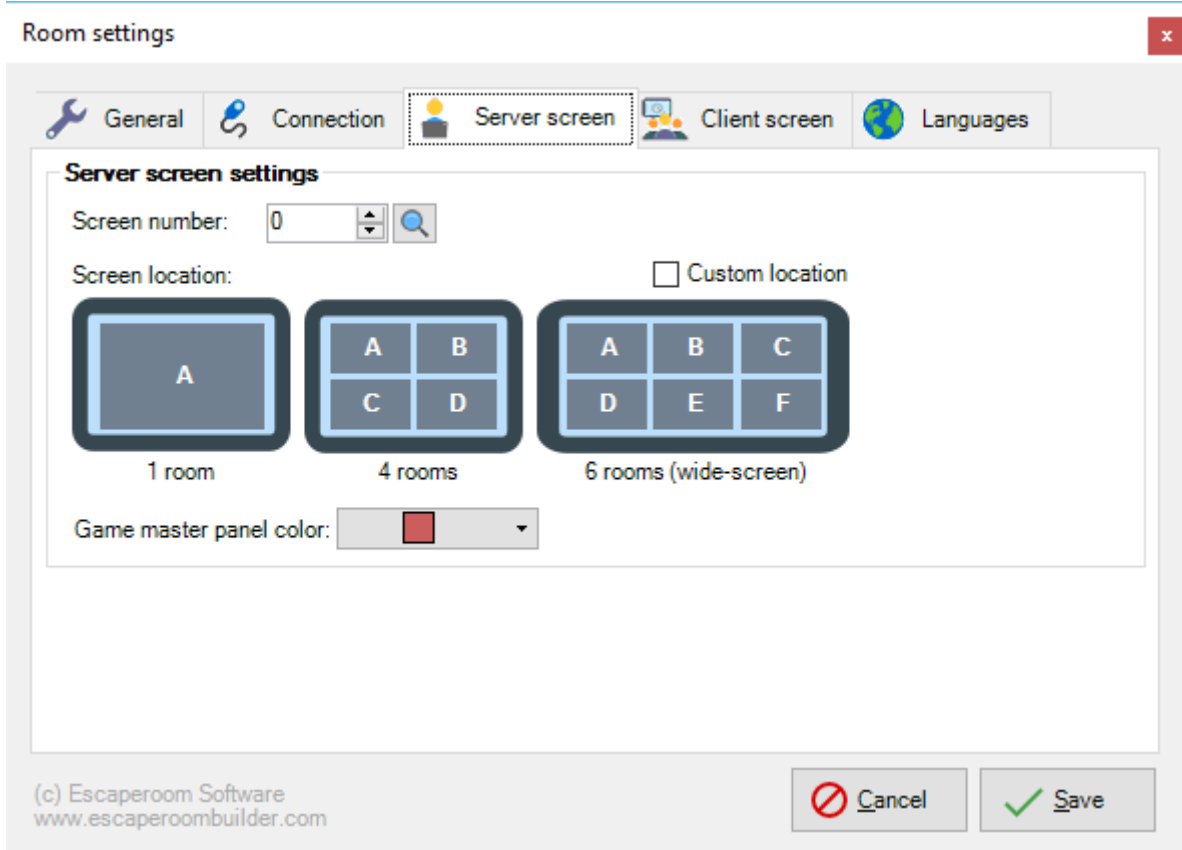


The screenshot shows a dialog box titled "External controller" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- ID:** A text input field containing the number "1". To its right is a checked checkbox labeled "Enabled".
- Description:** A text input field containing "New microcontroller".
- Connection:** A dropdown menu currently showing "Serial".
- Port:** A spinner control showing the number "1".
- At the bottom, there are two buttons: "Cancel" (with a red X icon) and "Save" (with a green checkmark icon).

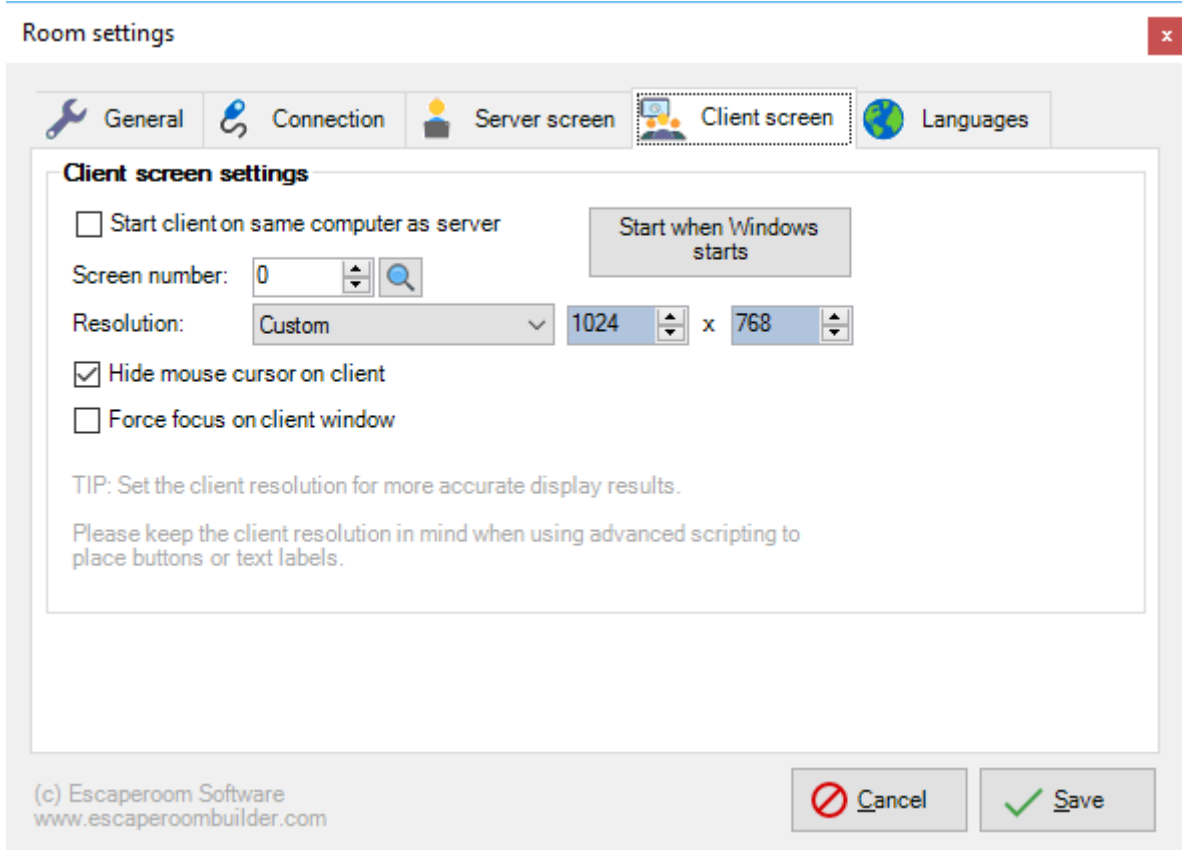
External controller	Details
ID	Use the ID to assign specific actions to an external controller in the wizard or in the script editor.
Description	Describe the task or location of this external controller for future reference.
Enabled	Enable or disable an external controller.
Connection	Connection types TCP and Serial are supported. Please check the included demo projects for code samples and Escape Room Software rooms.
Port	Enter the port number for the external connection.

### 5.5.7. Server screen



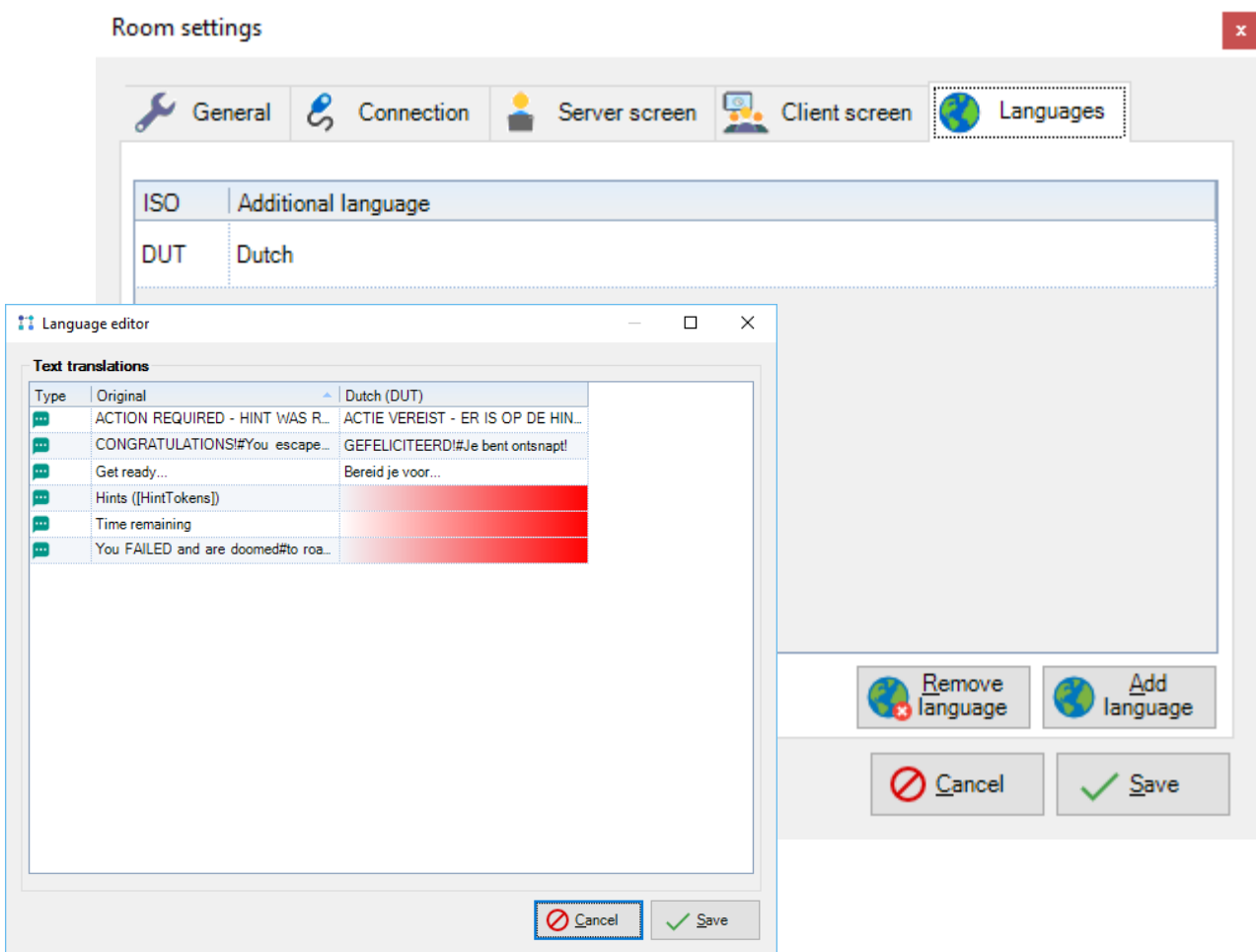
Server screen settings	Details
Screen number	Select a screen to show SERVER MODE on when working with multiple screens. Use the value 0 to not specify a specific screen.  After changing the screen, please reselect the screen location to update X, Y locations and sizes.
Identify screens	Press the magnifying glass button to identify the screen numbers (they show large red in the center of the respective screens)
Screen location	Use the screen location tool to quickly set a location and size to start when in SERVER MODE.  Use 4 or 6 room location presets when hosting multiple rooms on the same server computer.
Custom location (location X, Y, width, height)	You can also manually set the location and size fields if needed using actual coordinates and sizes.
Game master panel color	Shows the configured game master panel color for this room.

### 5.5.8. Client screen



Client screen settings	Details
Start client on same computer as server	When checked, a CLIENT MODE for this room is also started when starting SERVER MODE. Use this when you have 1 computer to run both the CLIENT and SERVER on.
Start when Windows starts	Toggle whether Windows automatically starts this room in CLIENT MODE when Windows starts. Use this on dedicated CLIENT computers.  Please note that administrator rights are required to use this function.
Screen number	Select a screen to show CLIENT MODE on when working with multiple screens. Use the value 0 to not specify a specific screen.
Resolution	Select the resolution for the CLIENT. Keep the client resolution in mind when using advanced functions to place buttons or move text labels.  TIP: Set the client resolution for more accurate display results. Please keep the client resolution in mind when using advanced scripting to place buttons or text labels.
Hide cursor on client screen	Hides the cursor on the client screen, making it more suitable for touchscreen (default this is turned ON).
Force focus on client window	Turn on if other software on a dedicated client computer causes the client application running there to lose focus (causing client keyboard or barcode input to fail). This feature checks and restores focus if needed every 2 seconds (default this is turned OFF).

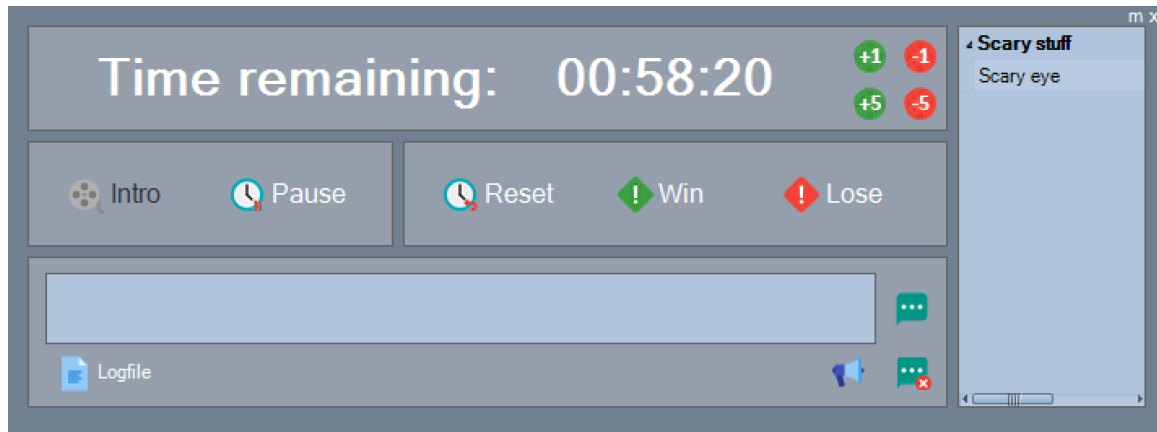
## 5.5.9. Languages



Language settings	Details
Add language	Add an additional supported language, on top of the usual language.
Remove language	Remove a language.

Language editor	Details
Grid	<p>The grid shows the original language and the alternate language of all text in the room file.</p> <p>Please note that untranslated lines will show red and will not be translated when running a game session.</p>

## 5.6. Game master panel



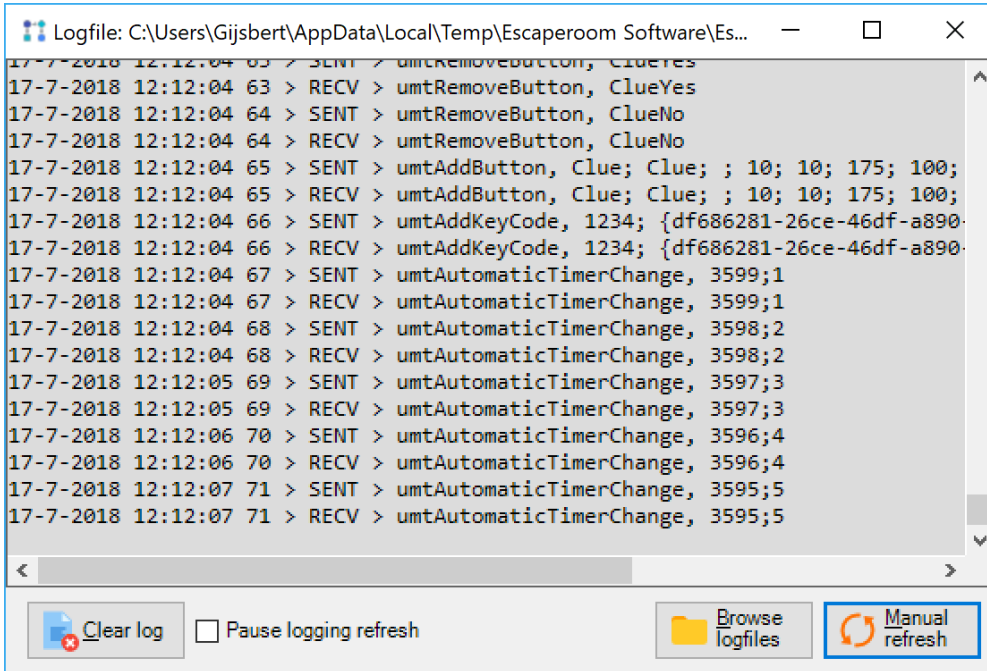
Game master panel	Details
Time remaining	Shows remaining time. Turns red when <10 minutes. When the timer expires, the room is automatically lost and runs the LOST script.
Time +/- buttons	Increase or decrease the remaining time with 1 or 5 minutes. An alert will be sent to client on each change.
m-button (red or white)	Indicates whether the room sound is muted. Click to toggle audio.
+ / - button	Increase or decrease size of GM panel for easier access to the clues list

Game panel	Details
Intro	Play intro. Runs the INTRO script.
Start / Pause	Start or Pause the room. Runs START script on initial start.
Reset	Reset the room.
Win	Instantly win the room. Runs the WIN script.
Lose	Lose the room. Runs the INSTALOSE script which kicks off the normal lose (time expired) script by default.

Message panel	Details
Message / send	Sends text message to the client.
Clear message	Clears all messages and images on the client.
Alert tone	Sends an ALERT to the client.
Logfile	Shows the logfile window.

Clues panel	Details
Clues list	Shows all manual clue actions available for this room grouped per category. Doubleclick to activate a manual clue action.  Useful for prepared clues and scary effects. The number of times you used specific clues is indicated, some clues may show as "stiked through" to indicate progress.

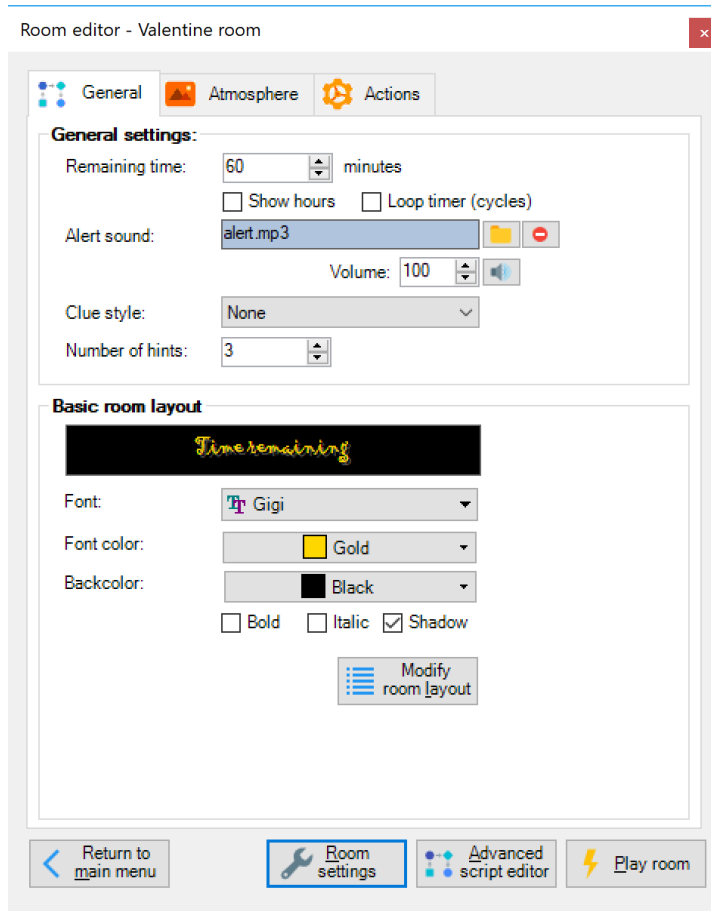
### 5.6.1. Logfile window



Name	Details
Log	<p>Displays the log results. Logfiles are saved in the Windows temporary folder. Every day a new logfile is saved.</p> <p>Errors will be shown in <b>RED</b>.            Home automation logs will be shown in <b>BLUE</b>.            Microcontroller logs will be shown in <b>PURPLE</b>.</p>
Manual refresh	Press the refresh button to see the most recent information.
Browse logfiles	Opens the Windows file explorer in the temporary logfile folder.
Pause logging refresh	Pause logging to be able to scroll through the log manually.
Clear log	While testing your room, you may want to clear Today's logfile now and then to avoid scrolling for recent info.

## 5.7. New room wizard dialog

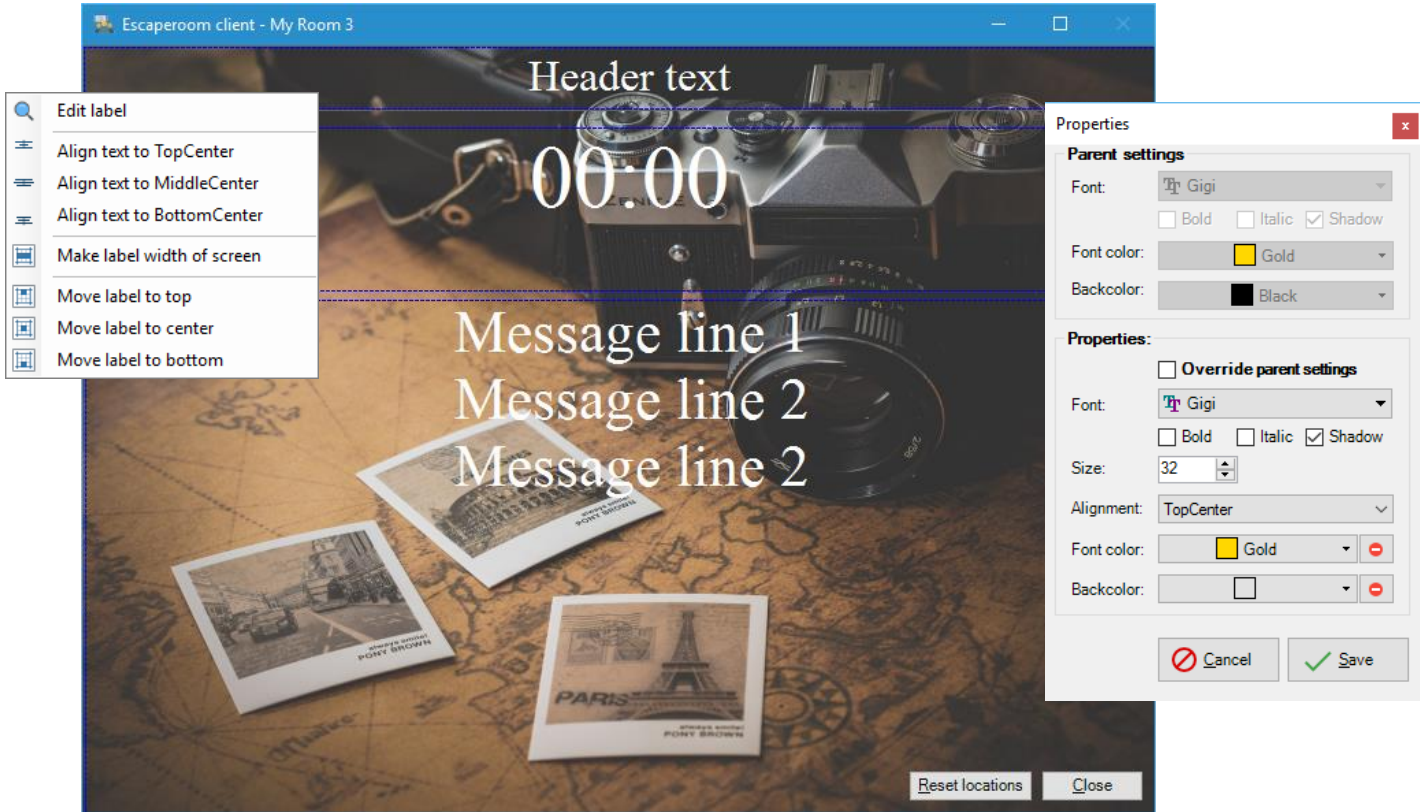
### 5.7.1. General



General settings	Details
Remaining time	Configure the initial remaining time for this room.
Show hours	Whether or not to show hours on the timer (default OFF)
Loop timer (cycles)	When timer finishes, instead of losing the game, the timer will restart. This enables you to create cycles.
Alert sound, volume	Configure a specific alert sound for this room.
Clue style	Select a style of hint button (none, button only, counter (decreasing with clue limit), counter increasing)
Number of clues	Max. number of hints when using a counter hint button.

Basic room layout	Details
Font	Select a main font for all labels.
Font color	Select a font color for all labels
Backcolor	Select a backcolor for the client screen
Bold, italic, shadow	Configure additional font properties
Preview room layout	Preview the game screen and set locations and sizes for labels. You can also override colors and font styles.

## 5.7.2. Modify room layout



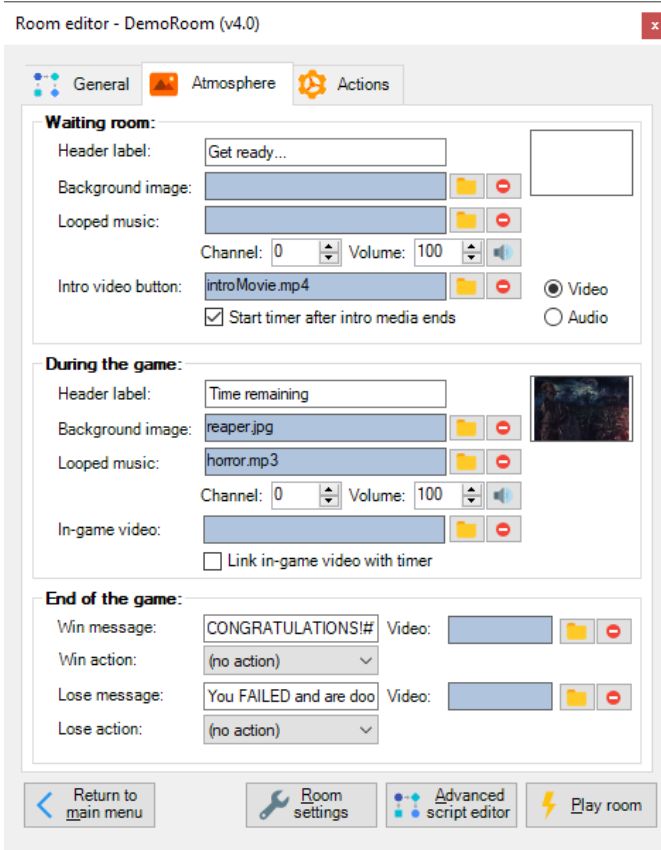
Basic room layout dialog	Details
Header label	This label shows a header caption during the game.
Timer label	This label shows the timer during the game.
Message label	This label shows messages during the game.
Reset locations	Reset the label locations on the screen to default.
Close	Save and close the basic room layout dialog.

Action	Mouse action	Keyboard action
Move label	Drag & drop	Arrow keys
Resize label	Resize using border edges	Increase width: <b>Home</b> , decrease width: <b>End</b> Increase height: <b>PgUp</b> , decrease height: <b>PgDown</b>

Context menu	Details
Label properties	Open the label properties dialog
Align label to...	Set the textalignment of the selected label.
Width of screen	Make the label the width of the entire screen.
Move label to...	Move the label to a preset location of the screen.

Properties dialog	Details
Parent settings	Every label adheres to the font settings set in the wizard.
Override parent settings	Enable to override the wizard font settings with custom settings for this label.
Font	Set the font family of the selected label.
Size	Set the fontsize of the selected label.
Alignment	Set the textalignment of the selected label.
Colors	Set the front and back colors of the selected label.

### 5.7.3. Atmosphere

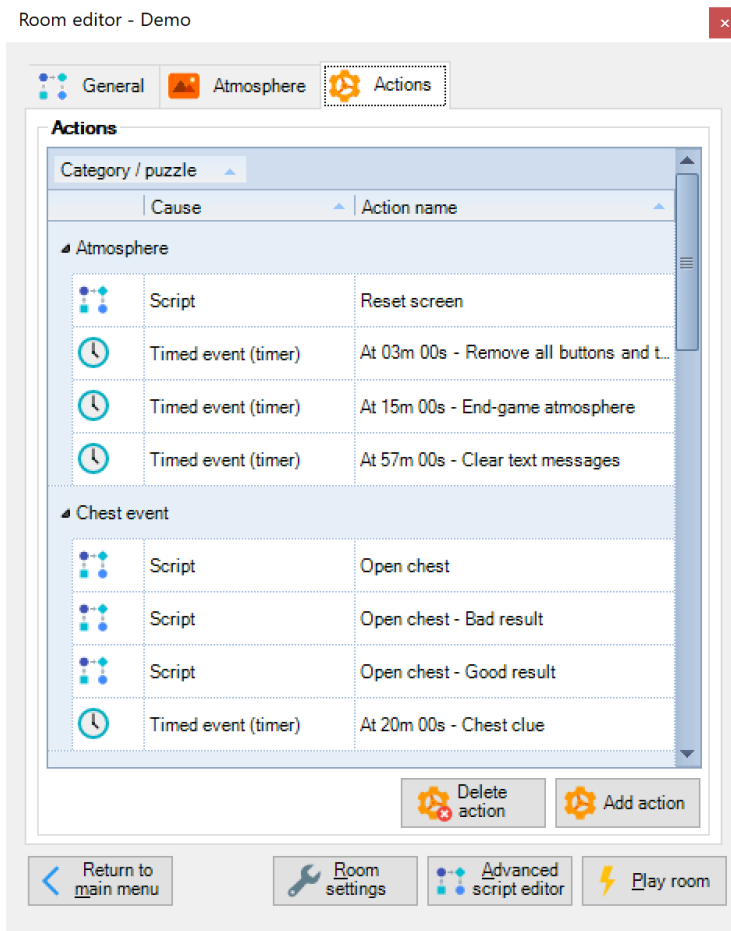


Waiting room	Details
Header label	Set the text of the header label before the game.
Background image	Set the background image before the game.
Looped music, volume, channel	Import audio to play looped in the waiting room. Channel 0 means NO CHANNEL and always play on a separate channel.
Intro video button	Import a video to play when pressing the Intro button.
Start timer after media ends	Automatically start the timer after the intro audio or video ends.

During the game	Details
Header label	Set the text of the header label during the game.
Background image	Set the background image before the game.
Looped music, volume, channel	Import audio to play looped during the game.
In-game video	Import a video to play during the game.
Link in-game video with room timer	Make the video jump along with manual or scripted timer manipulations.

End of the game	Details
Win message	Show a message when the game is won.
Win video	Import a video to play when the game is won.
Win action	Trigger an action upon Win condition.
Lose message	Show a message when the game is lost.
Lose video	Import a video to play when the game is lost.
Lose action	Trigger an action upon Lose condition.

## 5.7.4. Actions



### Actions

Actions grid and buttons

### Details

Manage various types of actions for this room.

## 5.8. Action dialog

### 5.8.1. Cause

The screenshot shows the 'Add action' dialog box with the following details:

- General:**
  - Action name:
  - Category/puzzle:
  - Categories/puzzles and actions are shown in alphabetical order on the Game Master panel.
- Cause:**
  - None (stand-alone script, triggered by other actions)
  - Clue (manually click on game master panel)**
  - Timed, when timer reaches...  min.  sec.
  - Timed, based on PC time schedule
  - Keypad entry
  - Home automation sensor

Please configure an effect on the next tabpage >>

GUID: {848e92d0-299c-454b-a799-3b4227ddb862}

General	Details
Action name	Name of the action will also be used as script filename. Example: "Puzzle #1 – Clue #2"
Category/puzzle	Divide your actions into categories. Clues are displayed in categories in the Game Master panel.
GUID	The action GUID is an identifier that can be used for scripting and incoming API calls

Cause	Details
None (stand-alone action, triggered by other actions)	This action may be called by other actions, or external actions via TCP connection.
Clue (manually click on game master panel)	Action is shown as a clues on the game master panel during the game. It must be triggered manually by the game master.
Timed, when timer reaches (min., sec.)	Action is triggered at a specific time in the game.
Timed, based on PC time schedule	Action is triggered at a specific time on the PC clock.
Keypad entry	Action is triggered when a valid keycode has been entered.
Home automation sensor	Action is triggered when a home automation sensor is triggered.

### 5.8.2. Timed, based on PC time schedule

**Timed, based on PC time schedule**

One time      Start date: 14- 7-2018 
  
 Date range      End date: 15- 7-2018 
  
 Monthly     
 Jan     Feb     Mar     Apr
  
 May     Jun     Jul     Aug
  
 Sep     Oct     Nov     Dec
  
  
Daily recurrence:   
 Sun     Mon     Tue     Wed
  
 Thu     Fri     Sat
  
  
Start time: 12:36:00 
  
 Repeat every hour until: 12:36:00

Timed, based on PC time schedule (optional)	Details
One time	Trigger this action only one time on a specific start date.
Date range	Trigger this action every day in the startdate/enddate daterange, you can also configure daily recurrence.
Monthly	Trigger this action monthly, according to daily recurrence at the proposed start time.
Start time	The start time of the action.
Repeat every hour until	Repeat this action every hour at the same minute as the start time for the rest of the day.

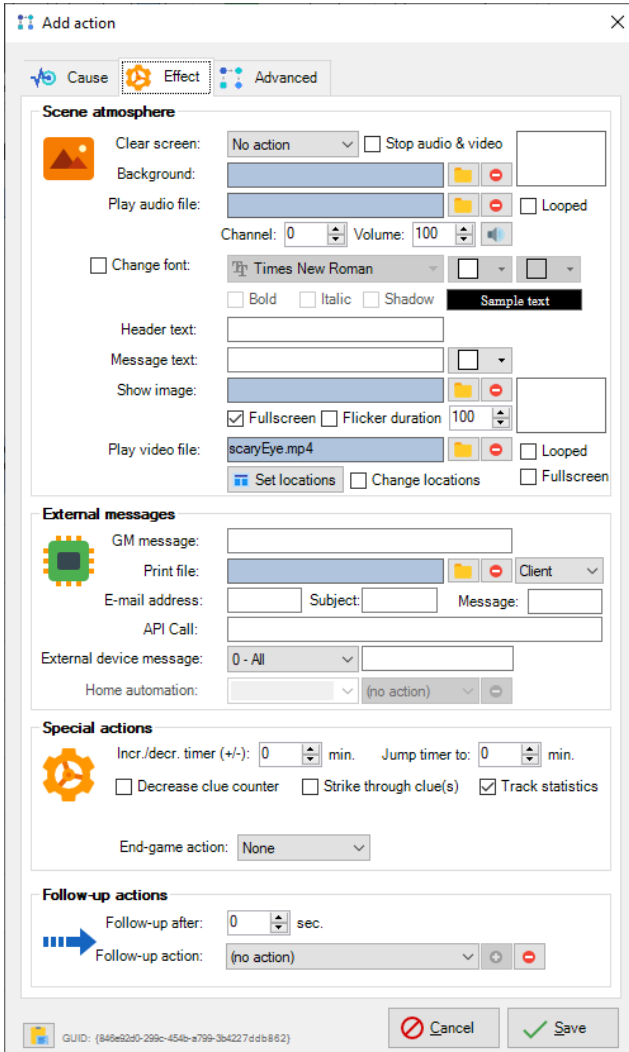
### 5.8.3. Keypad entry

**Keypad settings**

Keypad code: 
  
 Remove after correct entry
  
Wrong entry action:

Keypad settings (optional)	Details
Keypad code	A keypad code that triggers the current action. Please note that you can use an actual keyboard, keypad or barcode scanner to trigger an action like this.
Remove after correct entry	Whether or not a keypad code must be removed after correct entry.
Wrong entry action	Select an action that must be run when a wrong keycode is entered.

## 5.8.4. Effect



Action	Details
Clear screen	<i>No action, Clear all</i> or subcomponents on the screen
Background image	The shown image is used as background image.
Stop audio & video	
Play audio file, volume, channel	Play an audio file. Supported filetypes (.mp3, .wav)
Looped music	Stops earlier looped music and starts new looped music.
Font, weight, color	Change font, weight and colors of the main text.
Header text	Set header text.
Message text	Display message text.
Show image	Show an image. Supported filetypes (.jpg, .jpeg, .png, .bmp).
Full-screen image	The shown image is full-screen, overlapping text and timer.
Flickering image, flickers	Make the image flicker a number of times before disappearing.
Play video file	Play a video file. Supported filetypes (.avi, .mp4, .m4v, .mov)

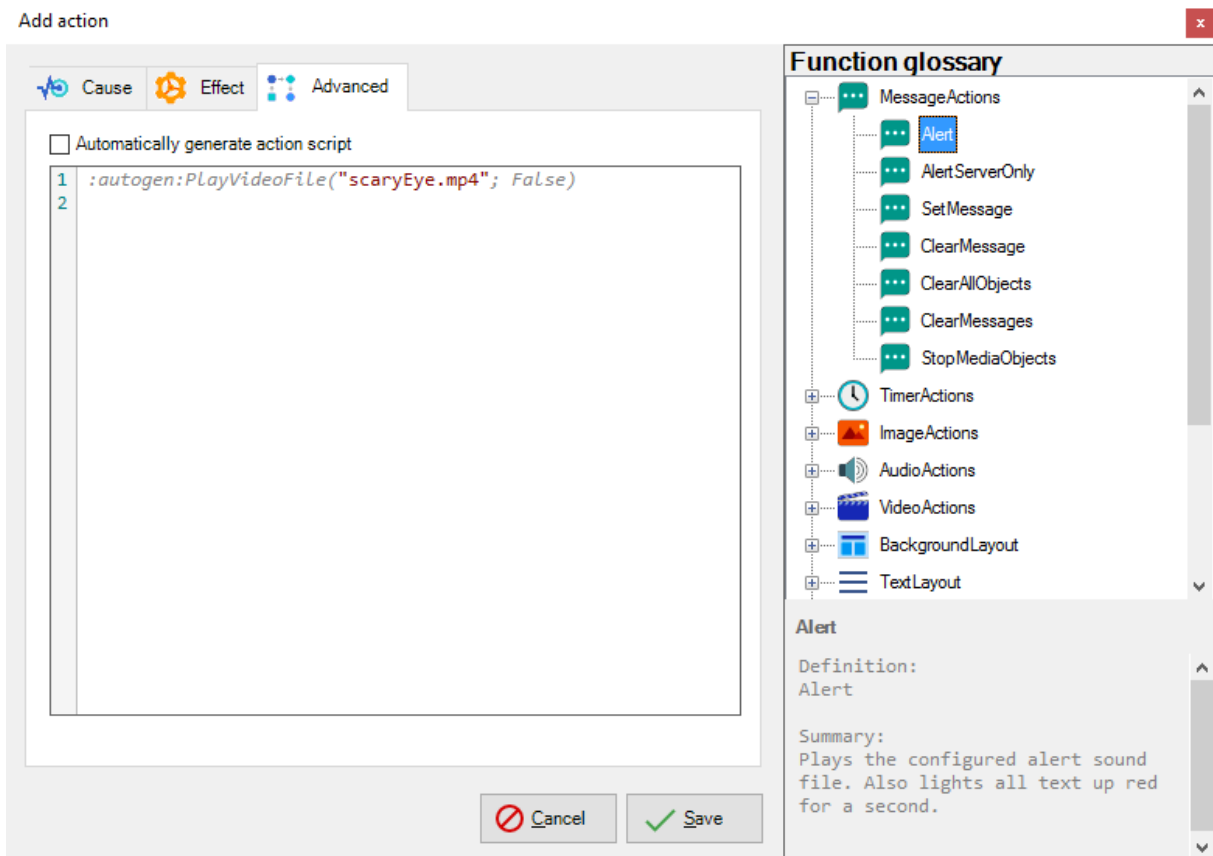
External messages	Details
GM message	Show a warning message to the Game Master (GM) only, with an accompanying ALERT sound. Use # as return character.
Print file, target	Print a PDF or Microsoft Word® file on the default printer of the client computer, the server computer or both.

E-mail message	Send an e-mail message with a specific subject to a target e-mail address. Please note that an SMTP server must be configured in the general settings.
Send API call	Runs API call on the SERVER computer. Use ? for params, & as param splitter and %20 for spaces.  Example: <a href="http://192.168.100.2:8080/someapi/runaction?param1=0&amp;param2=test%20me">http://192.168.100.2:8080/someapi/runaction?param1=0&amp;param2=test%20me</a>  You can do API calls on the Escaperoom Software self-hosted API. Be careful that the API call does not call itself, this will result in an endless loop and freeze the application.
External device message	Send a plain string message to one specific or all configured external devices (TCP, and serial connections are supported). You can use variable names using [brackets]
Home automation action	Trigger a home automation switch.

Special actions	Details
Change timer	Add/remove a number of minutes or seconds. 0=no action.
Jump to	Jump to a specific point in time. 0=no action.
Increase/decrease clue counter	Increase/decrease the clue counter variable by 1. (see "Clue button style")
Strike through clue(s)	Strike through this clue and all earlier strikethrough-enabled clues in this category/puzzle (earlier according to alphanumeric sortorder) on the Game Master panel. (only applies if clue is shown on the Game Master panel)
Strike through clue	This clue will be stricken through in the clue selection list.
Statistics: Track this script	A statistics counter keeps track of every time this script was run. See the chapter on <i>Statistics</i> in this manual for additional information.
Endgame actions	Set the game status to WIN GAME or LOSE GAME (runs the InstaLose script)

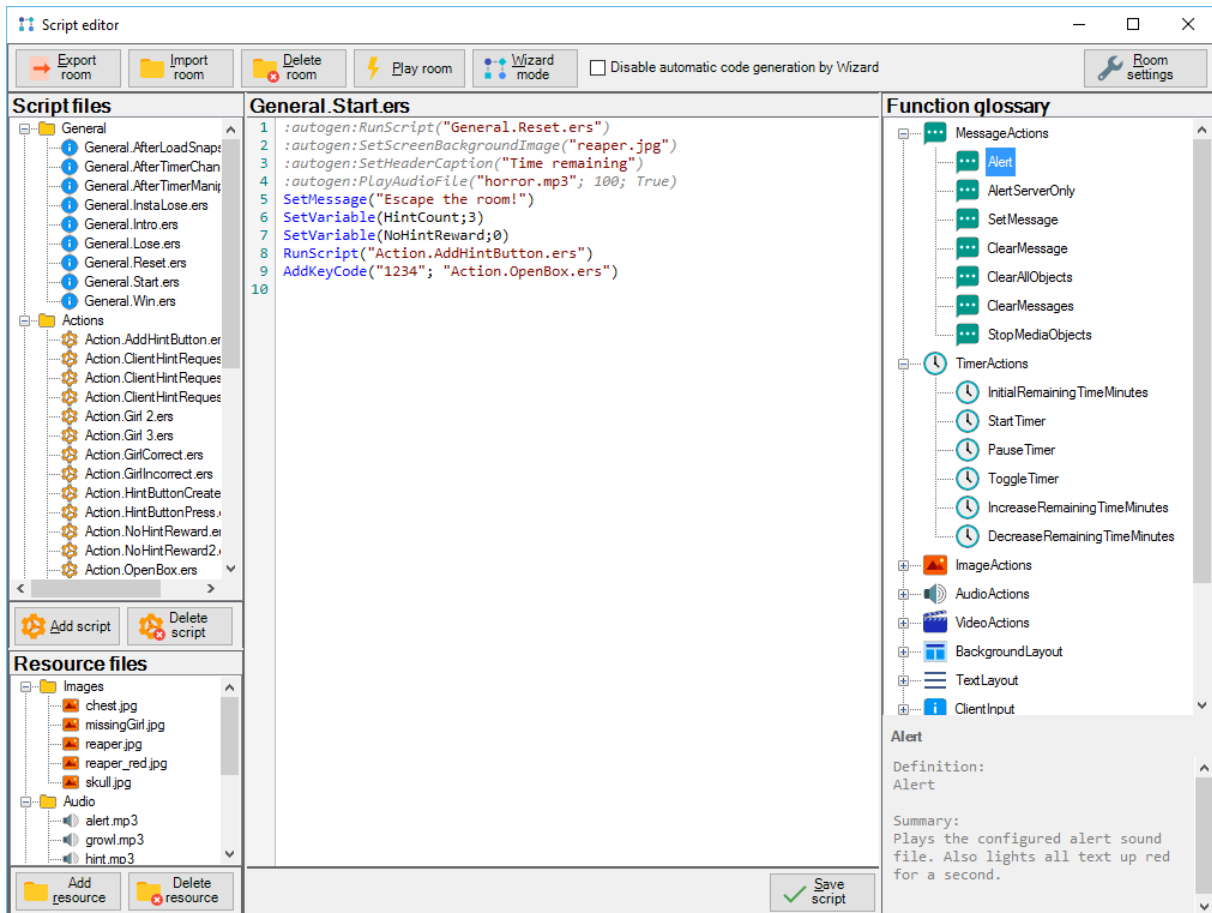
After action	Details
Follow-up after	Follow-up with another action (if configured) after a number of seconds, or immediately.  Please note that <b>actions probably shouldn't call themselves as a follow-up action.</b>
Follow-up action	Assign a follow-up action to this action.

## 5.8.5. Advanced



Advanced	Details
Automatically generate action script (default=ON)	<p>Toggle whether this action script is automatically generated. You can disable this check to manually modify the script.</p> <p>Please note that wizard functions for this script will be disabled when doing so.</p>
Script editor	<p>Shows the current selected script from the script files tree. When making a modification, a * is shown behind the script file in the script files tree.</p> <p>Functions are colored <b>blue</b>, strings are colored <b>red</b> for easier working.</p> <p>{purple text are GUIDs, which are unique identifier codes for other scripts. Hover above a GUID to see the actual name of the script that is referred to. Use the right mousebutton to insert a GUID for any existing script}</p> <p>//Remarkd tex is shown <b>green</b> and execution of these lines is skipped</p> <p>:autogen: code is <b>gray</b>. It is code autogenerated by the wizard and may be overwritten when revisiting wizard mode.</p>
Function glossary tree	Displays all functions. Doubleclick to add a function tot he script editor.
Function glossary info panel	Describes the syntax and details of the function selected in the function glossary tree.

## 5.9. Advanced script editor

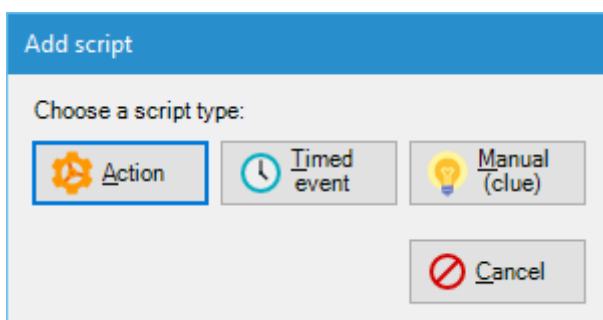


Room management panel	Details
Export room	Export the room folder to a ZIP file in the room zip folder (see the configuration screen).
Import room	Import a room ZIP file from the room zip folder (see the configuration screen).
Delete room	Delete the current room. You will be asked if you wish to create a new, empty room. The New room wizard will appear.
Play room	Start the room in DEBUG mode for easy testing.
Wizard mode	Quickly switch between the script editor and the wizard. Please note that the wizard autogenerates codes and usually, but not always respects manually added code.
Disable automatic code generation by wizard	Check this box to make sure the editor always starts in script editor mode instead of wizard mode.
Room settings	Open the room settings screen.

Files pane (left)	Details
Script files tree (general, actions, timed events, manual actions)	Shows all script files in the room folder.
New script button	Create a new script. You will be asked what type of script you wish to add. Please note that you cannot add general scripts, as they are a fixed set of scripts.
Delete script button	Delete the selected script from the room folder. Please note that you cannot delete general scripts, as they are a fixed set of scripts.

Rename script	When using the right mouse button on a scriptfile, you can rename it.
Resource files tree (images, audio, video, other)	Displays and categorises all resource files in the room folder. The right mousebutton als be used to access add/delete and previewing of the resource files.  Please note that you can also add resources by dragging files from your Windows explorer onto this tree.
Add resource	Add a resource tot the room folder.
Delete resource	Delete the selected resource from the room folder.

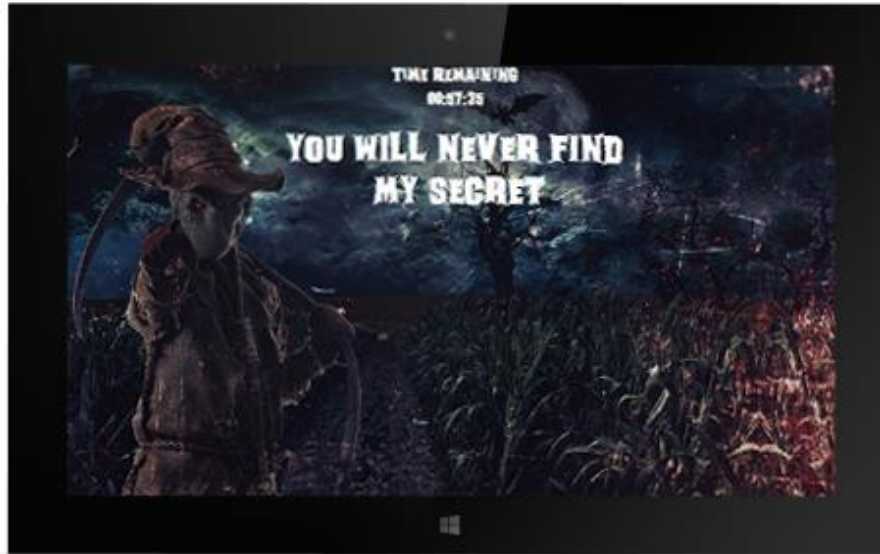
Script editor panel (center, right)	Details
Script editor	Shows the current selected script from the script files tree. When making a modification, a * is shown behind the script file in the script files tree.  Functions are colored <b>blue</b> , strings are colored <b>red</b> for easier working.  {purple text are GUIDs, which are unique identifier codes for other scripts. Hover above a GUID to see the actual name of the script that is referred to. Use the right mousebutton to insert a GUID for any existing script}  //Remarkd tex is shown <b>green</b> and execution of these lines is skipped  :autogen: code is <b>gray</b> . It is code autogenerated by the wizard and may be overwritten when revisiting wizard mode.
Save script button	Save the current opened script.
Function glossary tree	Displays all functions. Doubleclick to add a function tot he script editor.
Function glossary info panel	Describes the syntax and details of the function selected in the function glossary tree.



Add script	Details
Action	Add a generic action that can be kicked off by other scripts or triggers.
Timed event	Add a script that will fire at a specific time in the game (based on minutes and seconds).
Manual (clue)	Add a manual (clue) script that will show in the clue selector on the server screen.

### 5.10. In-room client screen

The client screen is *stay on top*.



Interaction possibilities	Details
Keyboard - F10 key	Toggle logfile visibility.
Keyboard - F11 key	Toggle normal/locked window.
Keyboard - F12 key	Close the client with the F12-key.
Touchscreen buttons	You can configure touchscreen buttons and associated scripts in the editor.
Keyboard/keypad input or USB barcode scanning	You can configure codes and associated scripts in the editor. Just enter the code in the keyboard and press ENTER.  Please note that a USB barcodescanner behaves just as a keyboard and should give a RETURN-character (ENTER key) at the end of scanning each barcode.

### 5.11. Limiting access on the client computer

Windows 8.x and 10 support Windows kiosk-mode. You can thoroughly lock the client screen computer and limit user access on the computer by running the application in kiosk mode. You do need a Pro version of Windows for this feature though.

Other tricks are to hide the taskbar, the charms panel in Windows 8.x or the Action Center in Windows 10.

Microsoft offers a number of solutions on their website.

### 5.12. Importing a room on a dedicated client computer

Use the main menu to import and configure a room when using a dedicated CLIENT computer.

- Make sure both CLIENT and SERVER run the same version of Escape Room Software
- Make sure both CLIENT and SERVER computers have power saving settings and screensavers disabled
- When using different fonts, make sure all CLIENT and SERVER computers have the proper fonts installed

## 6. Scripting guide

### 6.1. General

Every script that is executed runs all actions in chronological order. There is no pausing between actions.

### 6.2. Script types

The following script types are available:

Name	Purpose	Examples of use
General	General scripts are run at fixed core-events in the game.	See the next paragraph for more information.
Actions	These scripts are run only when referenced by another script.	Create reusable scripts that can be executed from multiple places. Also, a variable comparison could cause a specific action script to run.
Timed events	These scripts run automatically at the specific minute that you choose.	Increase tension by changing images and music in the last 10 minutes of play.
Manual actions	These scripts are available for the game master to run manually by doubleclicking on them.	Use manual actions for prepared clues or effects to influence the room ambiance.
Home automation scripts	These scripts are run when specific home automation triggers are fired.	Such scripts are fired when a home automation event is fired.

### 6.3. General script definitions

The general scripts have the following purpose:

Name	Event	Details
Reset	Automatically executed when the room is loaded, OR when the game master presses the RESET button.	Here you usually define the fonts, font sizes, colors, room timer value (default is 60 minutes), etc.
Intro	Executed when the game master presses the INTRO button.	Used for intro movies.
Start	Executed when the game master presses the START button.	
AfterTimerChange	Always executed after timer has changed	Use the ! variable to access the current time.
AfterTimerManipulation	Executed after time has been manipulated by manually pressing a + or - button on the server screen.	Use the ! variable to access the current time.  You could use this to match up a playing video with the room timer.
AfterManualMessage	Executed after the GameMaster (GM) manually sends a message using the button on the GM panel.	
Win	Executed when the game master presses the WIN button.	
Lose	Executed when the game timer expires.	
InstaLose	Executed when the game master presses the LOSE button.	Per default the InstaLose script automatically runs the normal General.Lose script.

## 6.4. Examples

### 1. How to create a room with a red background and white text? **(simple)**

*Modify script: General.Reset.ers*

```
SetScreenBackColor("red")
SetFontColor("white")
```

---

### 2. How to create a room with music that changes when there are 5 minutes remaining? **(simple)**

*Modify script: General.Start.ers*

```
PlayAudioFile("audiofile_normal.mp3"; 100; true)
```

*Add timed script at 5 minutes: Timed.005.ers*

```
PlayAudioFile("audiofile_scary.mp3"; 100; true)
```

---

### 3. How to create a room with a quiz using buttons that allows players to win time? **(advanced)**

*Modify script: General.Start.ers*

```
SetVariable(QuizAnswered;0)
```

*Add timed script at 45 minutes: Timed.045.ers*

```
SetImage("MysteriousPictureOfAGirl.jpg")
SetMessage("A girl from the village vanished#in this house..."; "white")
```

*Add timed script at 44 minutes: Timed.044.ers*

```
SetMessage("#Her#name#was#Emily"; "gray")
ClearMessages(10)
```

*Add timed script at 5 minutes: Timed.005.ers*

```
CompareVariable(QuizAnswered;=;0;"Action.Quiz.ers";"")
```

*Add action script: Action.Quiz.ers*

```
SetMessage("What was the name of the#missing girl?"; "yellow")
AddButton("Answer1"; "Rachel"; ""; C-300; 300; 250; 100; "GUID of Action GirlIncorrect")
AddButton("Answer2"; "Max"; ""; C+50; 300; 250; 100; "GUID of Action GirlIncorrect")
AddButton("Answer3"; "Emily"; ""; C-300; 500; 250; 100; "GUID of Action GirlCorrect")
AddButton("Answer4"; "Chloe"; ""; C+50; 500; 250; 100; "GUID of Action GirlIncorrect")
```

*Add action script: Action.GirlCorrect.ers*

```
SetVariable(QuizAnswered;1)
RemoveAllButtons
SetMessage("Correct!##You receive 3 minutes bonus time!"; "Limegreen")
IncreaseRemainingTimeMinutes(3)
ClearMessages(10)
```

*Add action script: Action.GirlIncorrect.ers*

```
SetVariable(QuizAnswered;1)
RemoveAllButtons
SetMessage("Alas, the answer is incorrect.")
ClearMessages(10)
```

---

Please import the included **SampleRoom** file to check out more examples.

## 7. Function glossary

### 7.1. MessageActions

Name	Sample code	Details
Alert	<code>Alert</code>	Plays the configured alert sound. Also lights all text up red for a second.
AlertServerOnly(string Message)	<code>AlertServerOnly("Notification message here!")</code>	<p>Shows a message in a red color on the server screen only, to inform the game master that something has happened or indicate that a manual action is required. Click on the label to hide it again. Also note that [variables] are supported.</p> <p>Also plays the configured alert sound, on the server even though it may be muted. If no special server sound is configured, the normal alert sound is played.</p>
SendMessage(string Message; optional string ColorName)	<code>SendMessage("Hello world!")</code> <code>SendMessage("Colored message here!"; "Red")</code> <code>SendMessage("This is a variable value: [variableName]")</code>	<p>Sends a text message. Use the color parameter to indicate a color. Use the # character for a line-break. Also note that [variables] are supported.</p> <p>Search the internet for .NET colornames table.</p>
ClearMessage	<code>ClearMessage</code>	Clears current shown message (if any).
ClearAllObjects(optional int Seconds)	<code>ClearAllObjects(0)</code>	Stops all playing media objects. Optionally after a number of seconds. Also clears message and any images.
ClearMessages(optional int Seconds)	<code>ClearMessages(0)</code>	Stops all playing media objects. Optionally after a number of seconds. Also clears message and any images.
StopMediaObjects(optional int Seconds)	<code>StopMediaObjects(0)</code>	Clears message and image objects. Optionally after a number of seconds.

## 7.2. TimerActions

Name	Sample code	Details
<code>InitialRemainingTimeMinutes(int Minutes)</code>	<code>InitialRemainingTimeMinutes(60)</code>	Sets initial remaining time for the room to a number of minutes.
<code>StartTimer</code>	<code>StartTimer</code>	Starts the timer.
<code>PauseTimer</code>	<code>PauseTimer</code>	Pauses the timer.
<code>ToggleTimer</code>	<code>ToggleTimer</code>	Toggles timer between started and paused.
<code>IncreaseRemainingTimeHours(int Hours)</code>	<code>IncreaseRemainingTimeHours(0)</code>	Increases game timer with a number of hours.
<code>IncreaseRemainingTimeMinutes(int Minutes)</code>	<code>IncreaseRemainingTimeMinutes(0)</code>	Increases game timer with a number of minutes.
<code>IncreaseRemainingTimeSeconds(int Seconds)</code>	<code>IncreaseRemainingTimeSeconds(0)</code>	Increases game timer with a number of seconds.
<code>DecreaseRemainingTimeHours(int Hours)</code>	<code>DecreaseRemainingTimeHours(0)</code>	Decreases game timer with a number of hours.
<code>DecreaseRemainingTimeMinutes(int Minutes)</code>	<code>DecreaseRemainingTimeMinutes(0)</code>	Decreases game timer with a number of minutes.
<code>DecreaseRemainingTimeSeconds(int Seconds)</code>	<code>DecreaseRemainingTimeSeconds(0)</code>	Decreases game timer with a number of seconds.
<code>JumpToTimer(int Minutes)</code>	<code>JumpToTimer(0)</code>	Jump the timer to a specific number of minutes.
<code>LoopTimer</code>	<code>LoopTimer</code>	Reset the time elapsed to 0 after timer finishes, without losing the game. This allows for an infinite cyclic strategy.

### 7.3. ImageActions

Name	Sample code	Details
<code>SetImage(string FileName; bool FullScreen)</code>	<code>SetImage("imagefile.jpg"; False)</code>	<p>Loads and shows image. Use True or False to indicate whether the image should be shown full screen.</p> <p>Supported filetypes (.jpg, .jpeg, .png, .bmp)</p>
<code>FlickerImage(int NumberOfFlickers)</code>	<code>FlickerImage(100)</code>	<p>Flickers the loaded image a number of times for a scary effect.</p>
<code>ClearImage</code>	<code>ClearImage</code>	<p>Hides current shown image (if any).</p>
<code>SetImageLocationSize(int Left; int Top; int Width; int Height)</code>	<code>SetImageLocationSize(10; 10; 10; 10)</code>	<p>Sets screen location and size for the IMAGE box.</p> <p>Use the C (=Center) character to offset from center (example: C-100 or C+100) for left and top variables.</p> <p>Use % character for width and height to make this object a specific percentage of the screen size, or to place the top or left at a specific percentage of the screen.</p>

## 7.4. AudioActions

Name	Sample code	Details
<code>PlayAudioFile(string FileName; int Volume; bool Loop; optional int Channel)</code>	<code>PlayAudioFile("audiofile.mp3"; 100; False)</code>	Plays audio file at a specified volume percentage. Use True or False to indicate whether the audio should be played in a loop. Multiple files can be played simultaneously. Optionally use different channels to organise audio.  Supported filetypes (.mp3, .wav)
<code>PlayAudioFileAndStartTimer(string FileName; int Volume; bool Loop; optional int Channel)</code>	<code>PlayAudioFileAndStartTimer("audiofile.mp3"; 100; False)</code>	Plays audio file at a specified volume percentage and starts the timer upon finishing.  Supported filetypes (.mp3, .wav)
<code>ChangeAudioVolume(int Volume; optional int Channel)</code>	<code>ChangeAudioVolume(100)</code>	Plays a running audio file at a specified volume percentage. Optionally use different channels to organise audio.
<code>StopAllAudio(optional int Channel)</code>	<code>StopAllAudio</code>	Stops all playing audio objects. Optionally choose a specific channel to stop audio on.
<code>SetAlertFile(string FileName; optional int Volume)</code>	<code>SetAlertFile("alert.mp3"; 100)</code>	Set audio file for playing alerts.
<code>SetServerAlertFile(string FileName; optional int Volume)</code>	<code>SetServerAlertFile("serverAlert.mp3"; 100)</code>	Override alert audio file on the server.

## 7.5. VideoActions

Name	Sample code	Details
PlayVideoFile(string FileName; bool Loop)	<code>PlayVideoFile("videofile.avi"; False)</code>	Plays video file. Use True or False to indicate whether the video should be played in a loop. Only one file can be played simultaneously.  Supported filetypes (.avi, .mp4, .m4v, .mov)
PlayVideoFileAndStartTimer(string FileName; bool Loop)	<code>PlayVideoFileAndStartTimer("videofile.avi"; False)</code>	Same as PlayVideoFile, but also starts timer upon finishing. Used as followup for the intro button.
SetVideoFileTimer(int Seconds)	<code>SetVideoFileTimer(!)</code>	Set the video timer to a specific time in seconds.  Use the ! variable to match up with the current timer value.
SetVideoFileTimer(special function)		
OffsetVideoFileTimer(int OffsetInSeconds)	<code>OffsetVideoFileTimer(0)</code>	Offset the video timer with a specific time in seconds. You can use both positive and negative values.
PauseVideoFile	<code>PauseVideoFile</code>	Pause current playing video file (if any).
ResumeVideoFile	<code>ResumeVideoFile</code>	Resume current playing video file (if any).
StopVideoFile	<code>StopVideoFile</code>	Stop and hide current playing video file (if any).
TimedStopVideoFile(int Seconds)	<code>TimedStopVideoFile(10)</code>	Stop and hide current playing video file (if any) after a number of seconds.
SetVideoLocationSize(int Left; int Top; int Width; int Height)	<code>SetVideoLocationSize(10; 10; 10; 10)</code>	Sets screen location and size for the VIDEO player.  Use the C (=Center) character to offset from center (example: C-100 or C+100) for left and top variables.  Use % character for width and height to make this object a specific percentage of the screen size, or to place the top or left at a specific percentage of the screen.

## 7.6. BackgroundLayout

Name	Sample code	Details
SetScreenBackColor(string ColorName)	<code>SetScreenBackColor("Black")</code>	Sets the screen bgcolor by color name. The default value is "black".  Search the internet for the .NET color names table.
SetScreenBackgroundImage(string FileName)	<code>SetScreenBackgroundImage("image.jpg")</code>	Loads and shows the lower backgroundimage. Make sure the background image matches the client screen resolution to avoid black bars on top or on the sides. Supported filetypes (.jpg, .jpeg, .png, .bmp)
ClearScreenBackgroundImage	<code>ClearScreenBackgroundImage</code>	Clears the lower background image of the screen.
SetClientScreenPower(bool)	<code>SetClientScreenPower(False)</code>	Turns client screen(s) on/off. Does not work when running client and server on the same computer.

## 7.7. TextLayout

Name	Sample code	Details
SetFontFamily(string FontFamily)	<code>SetFontFamily("Times New Roman")</code>	<p>Changes the font of the HEADER and MESSAGE and TIMER labels and BUTTONS. Make sure that the chosen font is installed on the server and on all client computers.</p> <p>The default value is "Times New Roman".</p>
SetFontBold(bool)	<code>SetFontBold(True)</code>	Sets the font.Bold property of all labels and buttons.
SetFontItalic(bool)	<code>SetFontItalic(True)</code>	Sets the font.Italic property of all labels and buttons.
SetFontShadow(bool)	<code>SetFontShadow(True)</code>	Sets the font.Shadow property of all labels and buttons.
SetFontColor(string ForeColorName; string BackColorName)	<code>SetFontColor("white"; "transparent")</code>	<p>Sets the font forecolor and backcolor for all labels and buttons by color name.</p> <p>The default value is "white".</p> <p>Search the internet for .NET colornames table.</p>
SetHeaderCaption(string Caption)	<code>SetHeaderCaption("Time remaining...")</code>	<p>Sets the value for the HEADER label. Useful for language translations.</p> <p>Also note that [variables] are supported.</p>
SetHeaderFontSize(int Size)	<code>SetHeaderFontSize(24)</code>	<p>Sets font size of the HEADER label.</p> <p>The default value is 24.</p>
SetHeaderFont(string FontFamily; int Size; bool Bold; bool Italic; bool DropShadow; string ForeColorName; string BackColorName)	<code>SetHeaderFont("Times New Roman"; 24; False; False; False; "white"; "transparent")</code>	<p>Sets font family, size, bold, italic, shadow and colors of the HEADER label.</p> <p>The default size value is 24.</p>
SetHeaderLocationSize(int Left; int Top; int Width; int Height; string TextAlignment)	<code>SetHeaderLocationSize(10; 10; 10; 10; Center)</code>	<p>Sets screen Location, Size and TextAlignment for the HEADER label.</p> <p>TextAlignment can be either Center, Left or Right. You can also use TopLeft, TopCenter, TopRight, MiddleLeft, MiddleCenter, MiddleRight, BottomLeft, BottomCenter and BottomRight for more specific placement in the label.</p> <p>Use the C (=Center) character to offset from center (example: C-100 or C+100) for left and top variables.</p> <p>Use % character for width and height to make this object a specific</p>

		percentage of the screen size, or to place the top or left at a specific percentage of the screen.
TimerVisible(bool)	<code>TimerVisible(False)</code>	Shows or hides the TIMER label. You may want to use this when using a videofile that has the timer incorporated.
TimerFormat(string Format)	<code>TimerFormat("HHMMSS")</code>	Use the following masks: - HHMMSS = 00:00:00 (hours, minutes, seconds) - MMSS = 00:00 (minutes, seconds)
SetTimerFontSize(int Size)	<code>SetTimerFontSize(32)</code>	Sets font size of the TIMER label.  The default value is 32.
SetTimerFont(string FontFamily; int Size; bool Bold; bool Italic; bool DropShadow; string ForeColorName; string BackColorName)	<code>SetTimerFont("Times New Roman"; 32; False; False; False; "white"; "transparent")</code>	Sets font family, size, bold, italic, shadow and colors of the TIMER label.  The default size value is 32.
SetTimerLocationSize(int Left; int Top; int Width; int Height; string TextAlignment)	<code>SetTimerLocationSize(10; 10; 10; 10; Center)</code>	Sets screen Location, Size and TextAlignment for the TIMER label.  TextAlignment can be either Center, Left or Right. You can also use TopLeft, TopCenter, TopRight, MiddleLeft, MiddleCenter, MiddleRight, BottomLeft, BottomCenter and BottomRight for more specific placement in the label.  Use the C (=Center) character to offset from center (example: C-100 or C+100) for left and top variables.  Use % character for width and height to make this object a specific percentage of the screen size, or to place the top or left at a specific percentage of the screen.
SetSubtimerFontSize(int Size)	<code>SetSubtimerFontSize(32)</code>	Sets font size of the SUBTIMER label.  The default value is 32.
SetSubtimerFont(string FontFamily; int Size; bool Bold; bool Italic; bool DropShadow; string ForeColorName; string BackColorName)	<code>SetSubtimerFont("Times New Roman"; 32; False; False; False; "white"; "transparent")</code>	Sets font family, size, bold, italic, shadow and colors of the SUBTIMER label.  The default size value is 32.
SetSubtimerLocationSize(int Left; int Top; int Width; int Height; string TextAlignment)	<code>SetSubtimerLocationSize(10; 10; 10; 10; Center)</code>	Sets screen Location, Size and TextAlignment for the SUBTIMER label.  TextAlignment can be either Center, Left or Right. You can also use TopLeft, TopCenter, TopRight, MiddleLeft,

		<p>MiddleCenter, MiddleRight, BottomLeft, BottomCenter and BottomRight for more specific placement in the label.</p> <p>Use the C (=Center) character to offset from center (example: C-100 or C+100) for left and top variables.</p> <p>Use % character for width and height to make this object a specific percentage of the screen size, or to place the top or left at a specific percentage of the screen.</p>
<pre>SetMessageFontSize(int Size)</pre>	<pre>SetMessageFontSize(48)</pre>	<p>Sets font size of the MESSAGE label.</p> <p>The default value is 48.</p>
<pre>SetMessageFont(string FontFamily; int Size; bool Bold; bool Italic; bool DropShadow; string ForeColorName; string BackColorName)</pre>	<pre>SetMessageFont("Times New Roman"; 48; False; False; False; "white"; "transparent")</pre>	<p>Sets font family, size, bold, italic, shadow and colors of the MESSAGE label.</p> <p>The default size value is 48.</p>
<pre>SetMessageLocationSize(int Left; int Top; int Width; int Height; string TextAlignment)</pre>	<pre>SetMessageLocationSize(10; 10; 10; 10; Center)</pre>	<p>Sets screen Location, Size and TextAlignment for the MESSAGE label.</p> <p>TextAlignment can be either Center, Left or Right. You can also use TopLeft, TopCenter, TopRight, MiddleLeft, MiddleCenter, MiddleRight, BottomLeft, BottomCenter and BottomRight for more specific placement in the label.</p> <p>Use the C (=Center) character to offset from center (example: C-100 or C+100) for left and top variables.</p> <p>Use % character for width and height to make this object a specific percentage of the screen size, or to place the top or left at a specific percentage of the screen.</p>
<pre>SetAlertServerOnlyLocationSize(int Left; int Top; int Width; int Height; string TextAlignment)</pre>	<pre>SetAlertServerOnlyLocationSize(10; 10; 10; 10; Center)</pre>	<p>Sets screen Location, Size and TextAlignment for the ALERT SERVER ONLY label.</p> <p>TextAlignment can be either Center, Left or Right. You can also use TopLeft, TopCenter, TopRight, MiddleLeft, MiddleCenter, MiddleRight, BottomLeft, BottomCenter and BottomRight for more specific placement in the label.</p>

Use the C (=Center) character to offset from center (example: C-100 or C+100) for left and top variables.

Use % character for width and height to make this object a specific percentage of the screen size, or to place the top or left at a specific percentage of the screen.

## 7.8. ClientInput

Name	Sample code	Details
AddButton(string ButtonName; string Caption; string ImageFileName; int Left; int Top; int Width; int Height; string ScriptGUID)	<code>AddButton("NewButton"; "Caption"; "image.jpg (leave blank if not needed)"; 10; 10; 250; 100; "scriptGUID")</code>	<p>Adds a button to the client screen that kicks off a script when pressed.</p> <p>An example of useful interaction is for example a HINT button.</p> <p>Use the C (=Center) character to offset from center (example: C-100 or C+100) for left and top variables.</p> <p>Also note that [variables] are supported for the caption.</p> <p>Use % character for width and height to make this object a specific percentage of the screen size, or to place the top or left at a specific percentage of the screen.</p>
RemoveButton(string ButtonName)	<code>RemoveButton("Button1")</code>	Removes specific button from the client screen.
RemoveAllButtons	<code>RemoveAllButtons</code>	Removes all buttons from the client screen.
SetButtonFont(string FontFamily; int Size; bool Bold; bool Italic; bool DropShadow; string ForeColorName; string BackColorName)	<code>SetButtonFont("Times New Roman"; 24; False; False; False; "white"; "transparent")</code>	<p>Sets font family, size, bold, italic, shadow and colors for any BUTTONS placed subsequently.</p> <p>The default size value is 24.</p>
AddKeyCode(string KeyCode; string ScriptGUID)	<code>AddKeyCode("12345"; "scriptGUID")</code>	<p>Players can use a numpad or a barcodescanner attached to the client to enter codes or scan codes clues, triggering a server-side script.</p> <p>A barcode scanner automatically enters the code and sends the ENTER key, just like you would using a manual keyboard. Some barcode scanners are configured to also send a checkdigit at the end, so be sure to test for this.</p> <p>Please note that QR code scanning using webcam is not supported.</p>
RemoveKeyCode(string KeyCode)	<code>RemoveKeyCode("12345")</code>	Removes a specific keycode.
RemoveAllKeyCodes	<code>RemoveAllKeyCodes</code>	Removes all keycodes.
SetKeyCodeNotFoundScript(string ScriptGUID)	<code>SetKeyCodeNotFoundScript("12345"; "scriptGUID")</code>	Set a script to be fired when a wrong keycode is entered.
ClearKeyCodeNotFoundScript	<code>ClearKeyCodeNotFoundScript</code>	Clears the script to be fired when a wrong keycode is entered.

## 7.9. Scripting

Name	Sample code	Details
RunScript(string ScriptGUID)	<code>RunScript("scriptGUID")</code>	Runs another script. The other script is finished before the running script continues.
RunTimedScript(string TimerName; string ScriptGUID; int Seconds)	<code>RunTimedScript("TimerName"; "scriptGUID"; 0)</code>	Runs another script after a number of seconds. The running script continues in the mean time. Please note that you can run multiple scripts like this simultaneously.
StopTimedScript(string TimerName)	<code>StopTimedScript("TimerName")</code>	Stops the running a RunTimedScript action.
StopAllTimedScripts	<code>StopAllTimedScripts</code>	Stops the running all RunTimedScript actions.
WinRoom	<code>WinRoom</code>	Finish the game – Win status
LoseRoom	<code>LoseRoom</code>	Finish the game – Lose status

## 7.10. Connectivity

Name	Sample code	Details
ShellOnServer(string FileName; optional string Arguments)	<code>ShellOnServer("c:\notepad.exe"; /p)</code>	Runs an external command on the SERVER computer. Optionally add a commandline parameter.
ShellOnClient(string FileName; optional string Arguments)	<code>ShellOnClient("c:\notepad.exe"; /p)</code>	Runs an external command on the CLIENT computer. Optionally add a commandline parameter.
BroadcastSimpleCommand(string)	<code>BroadcastSimpleCommand("TEST1234")</code>	Broadcast a simple command over the configured main port. This can be used to notify other applications or devices listening on the same port.
BroadcastExternalControllerCommand(optional int ID; string value)	<code>BroadcastExternalControllerCommand("TEST1234")</code> or <code>BroadcastExternalControllerCommand(2, "TEST1234")</code>	Broadcast a simple command over the CONFIGURED PORT for all or a specific EXTERNAL DEVICE (low traffic, recommended when using external controllers). This can be used to notify external controller devices listening on the assigned External Controller TCP or serial port.  You can use variable names using [brackets].
APICall(string URL)	<code>APICall("http://192.168.100.2:8080/someapi/runaction?param1=0&amp;param2=test%20me")</code>	Runs API call on the SERVER computer.  Use ? for params, & as param splitter and %20 for spaces.
AdvancedAPICall(string ActionGUID)	<code>AdvancedAPICall("{000}")</code>	Runs advanced API call on the SERVER computer. Call can only be configured in the assigned action, not in a script.
PrintDocumentOnServer(string FileName)	<code>PrintDocumentOnServer("FileName.pdf")</code>	Print a document on the client computer default printer.  Supported filetypes (.pdf)
PrintDocumentOnClient(string FileName)	<code>PrintDocumentOnClient("FileName.pdf")</code>	Print a document on the client computer default printer.  Supported filetypes (.pdf)
SendSMTPEmailOnServer(string EmailAddress; string Subject; string Body; optional string AttachmentFileName)	<code>SendSMTPEmailOnServer("info@info.com"; "Subject line"; "Body lines")</code>	Send an email from the server computer.  Please note that you must configure an SMTP server in the settings.
ShutDownClientComputer	<code>ShutDownClientComputer</code>	Turns client computer off. Does not work when running client and server on the same computer.
RestartClientComputer	<code>RestartClientComputer</code>	Restarts client computer. Does not work when running client and server on the same computer.

## 7.11. HomeAutomation

Name	Sample code	Details
HaSetSwitch(string DeviceID; bool State)	<code>HaSetSwitch("1234"); True)</code>	Turn a Home Automation switch on or off.

Please note that home automation integration must be enabled in the room settings.

## 7.12. Variables

Name	Sample code	Details
SetVariable(variableName; var Value)	<a href="#">SetVariable</a> (variableName; 0)	Sets a variable in memory. A variant can be either a <i>string</i> or an <i>integer</i> value.
IncreaseVariable(variableName; int Value)	<a href="#">IncreaseVariable</a> (variableName; 1)	Increases an integer variable with a specific value.
DecreaseVariable(variableName; int Value)	<a href="#">DecreaseVariable</a> (variableName; 1)	Decreases an integer variable with a specific value.
CompareVariable(variableName; operator; variableName; string ScriptGUIDWhenTrue; string ScriptGUIDWhenFalse)	<a href="#">CompareVariable</a> (variableName;=;variableName;"scriptGUIDWhenTrue";"scriptGUIDWhenFalse")	Compares two variables. Assign scripts to run in case of true and false. Leave empty to not run a script.  Use =, <>, < or > as operators for comparison.  Use the ? character as a variable for a random result of 0 or 1. Use the ! character to get the elapsed time in seconds.
SaveVariableToSessionStats(variableName; int CustomFieldSlot)	<a href="#">SaveVariableToSessionStats</a> (variableName; 1)	Save a variable to one of the 9 custom fields in the current session statistics record.  Please note that there are 1-9 custom field slots available.
SaveScriptCountToSessionStats	<a href="#">SaveScriptCountToSessionStats</a>	Keep track of how many times this script was run.
StrikeThroughClues	<a href="#">StrikeThroughClues</a>	Strike through this clue and all earlier strike through-enabled clues in this category/puzzle on the Game Master panel.

Please note that you can place variables within [ and ] in message strings.

### 7.13. Reserved

Name	Sample code	Details
Unknown	n/a	Reserved function, do not use.
RestartClient	n/a	Reserved function, do not use.
AutomaticTimerChange	n/a	Reserved function, do not use.
PacketDropped	n/a	Reserved function, do not use.
UpdateClientVariables	n/a	Reserved function, do not use.
AskServerApplicationVersion	n/a	Reserved function, do not use.
ReturnServerApplicationVersion	n/a	Reserved function, do not use.
AskServerRoomLastChangedDate	n/a	Reserved function, do not use.
ReturnServerRoomLastChangedDate	n/a	Reserved function, do not use.

### 7.14. System variables

These are the system variables available in scripting mode:

Variable	Value type	Details	Can be used in functions:
!	int	Seconds elapsed	CompareVariable, SetVideoFileTimer
?	int	Random number 1 or 0	CompareVariable
&	string	Current language ISO	CompareVariable
[ ]	string	Use brackets to show a variable value	All string functions; SetMessage, etc.
[LastKeyCode]	string	Displays the last keycode entered by the players	All string functions; SetMessage, etc.
[HintTokens]	string	Displays the Clue Count	All string functions; SetMessage, etc.
^	string	Time taken in minutes, based on timerformat (HH:MM:SS or MM:SS)	All string functions; SetMessage, etc.
#	String	Return character	SetMessage

### 7.15. Incoming messages from external devices

If you send message from another device (TCP, UDP, Serial) using the proper syntax, the server runs a script of your choosing.

**Syntax example:** *@scriptGUID*

The @ character indicates that a script should be run, followed by the GUID of the script. This allows for external applications or devices to kick off scripts within the Escape Room Software server.












































































































































An external controller can add a variable by placing it between brackets after the normal syntax.

**Syntax example:** *@scriptGUID[variableName;Value]*

Before the script is executed on the server, first the variable of that name is set to the proposed value. If the variable does not yet exist, it is created.

## 8. NET colors overview

### 8.1. Color chart

 AliceBlue	 DarkTurquoise	 LightSeaGreen	 PapayaWhip
 AntiqueWhite	 DarkViolet	 LightSkyBlue	 PeachPuff
 Aqua	 DeepPink	 LightSlateGray	 Peru
 Aquamarine	 DeepSkyBlue	 LightSteelBlue	 Pink
 Azure	 DimGray	 LightYellow	 Plum
 Beige	 DodgerBlue	 Lime	 PowderBlue
 Bisque	 Firebrick	 LimeGreen	 Purple
 Black	 FloralWhite	 Linen	 Red
 BlanchedAlmond	 ForestGreen	 Magenta	 RosyBrown
 Blue	 Fuchsia	 Maroon	 RoyalBlue
 BlueViolet	 Gainsboro	 MediumAquamarine	 SaddleBrown
 Brown	 GhostWhite	 MediumBlue	 Salmon
 BurlyWood	 Gold	 MediumOrchid	 SandyBrown
 CadetBlue	 Goldenrod	 MediumPurple	 SeaGreen
 Chartreuse	 Gray	 MediumSeaGreen	 SeaShell
 Chocolate	 Green	 MediumSlateBlue	 Sienna
 Coral	 GreenYellow	 MediumSpringGreen	 Silver
 CornflowerBlue	 Honeydew	 MediumTurquoise	 SkyBlue
 CornsilK	 HotPink	 MediumVioletRed	 SlateBlue
 Crimson	 IndianRed	 MidnightBlue	 SlateGray
 Cyan	 Indigo	 MintCream	 Snow
 DarkBlue	 Ivory	 MistyRose	 SpringGreen
 DarkCyan	 Khaki	 Moccasin	 SteelBlue
 DarkGoldenrod	 Lavender	 NavajoWhite	 Tan
 DarkGray	 LavenderBlush	 Navy	 Teal
 DarkGreen	 LawnGreen	 OldLace	 Thistle
 DarkKhaki	 LemonChiffon	 Olive	 Tomato
 DarkMagenta	 LightBlue	 OliveDrab	 Transparent
 DarkOliveGreen	 LightCoral	 Orange	 Turquoise
 DarkOrange	 LightCyan	 OrangeRed	 Violet
 DarkOrchid	 LightGoldenrodYellow	 Orchid	 Wheat
 DarkRed	 LightGray	 PaleGoldenrod	 White
 DarkSalmon	 LightGreen	 PaleGreen	 WhiteSmoke
 DarkSeaGreen	 LightPink	 PaleTurquoise	 Yellow
 DarkSlateBlue	 LightSalmon	 PaleVioletRed	 YellowGreen
 DarkSlateGray			

## 9. How to connect to a micro controller

### 9.1. Arduino

#### 9.1.1. API connection

##### 9.1.1.1. Send API call from Arduino to ERS Server

```
#include <WiFi.h> // Use <ESP8266WiFi.h> for ESP8266

const char* ssid = "YOUR_SSID"; // Replace with your network SSID
const char* password = "YOUR_PASSWORD"; // Replace with your network password

WiFiClient client;

void setup() {
  Serial.begin(115200);
  delay(10);

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi");

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("\nConnected to WiFi");
}

void loop() {
  String ipAddress = "127.0.0.1"; // Change as needed; localhost may not work; use actual IP address when running locally
  int port = 8080; // Change as needed
  String roomId = ""; // Set the roomId if necessary
  String action = "{846e92d0-299c-454b-a799-3b4227ddb862}"; // Set the action as needed
  String url = "http://" + ipAddress + ":" + String(port) + "/ersapi/runaction?roomId=" + roomId + "&action=" + action;

  if (client.connect(ipAddress.c_str(), port)) {
    Serial.println("Connected to server");

    // Send HTTP GET request
    client.print(String("GET ") + url + " HTTP/1.1\r\n" +
      "Host: " + ipAddress + "\r\n" +
      "Connection: close\r\n\r\n");

    // Wait for server response
    while (client.connected() || client.available()) {
      if (client.available()) {
        String line = client.readStringUntil('\n');
        Serial.println(line);
      }
    }

    client.stop();
    Serial.println("Disconnected from server");
  } else {
    Serial.println("Connection failed");
  }
  delay(10000); // Wait 10 seconds before next request
}
```

### 9.1.1.2. Send API call from ERS Server to Arduino

```
#include <WiFi.h> // Use <ESP8266WiFi.h> for ESP8266
#include <WebServer.h>

// Wi-Fi credentials
const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASSWORD";

// Configurable IP address and port
IPAddress localIP(192, 168, 1, 100); // Set your fixed IP address
const int port = 80; // Set your desired port

WebServer server(port);

void setup() {
  Serial.begin(115200);

  // Connect to Wi-Fi
  WiFi.mode(WIFI_STA);
  WiFi.config(localIP, WiFi.gatewayIP(), WiFi.subnetMask());
  WiFi.begin(ssid, password);

  // Wait for connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("Connected to WiFi");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());

  // Define the API route
  server.on("/Ping", HTTP_POST, handlePing);

  // Start the server
  server.begin();
  Serial.println("Server started");
}

void loop() {
  server.handleClient();
}

void handlePing() {
  if (server.hasArg("plain")) {
    String message = server.arg("plain");
    Serial.print("Received: ");
    Serial.println(message);
    server.send(200, "text/plain", "Pong");
  } else {
    server.send(400, "text/plain", "Bad Request");
  }
}
```

### 9.1.2. TCP ethernet connection (not recommended, please use API)

```
// Escaperoom Software Arduino demo project TCP.Ethernet
// Version 1.6 (8/15/2017)
// Sample compiled from various sources found on www.arduino.cc
// Modified by Escaperoom Software

#include <Ethernet.h>
#include <SPI.h>

// please configure these settings
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // MAC address (get from sticker on device, or make up your own: must
be unique in the network)
byte ipLocal[] = { 192, 168, 100, 129 }; // local IP (must be unique in the network)
byte ipServer[] = { 192, 168, 100, 185 }; // server IP
unsigned int port = 13809; // room TCP port (the accompanying demo room sample uses TCP port 13009) Please do not
confuse the TCP port and sample with the UDP port and sample
bool isDebugMode = true;

// internal variables (don't change)
bool isConnecting = false;
bool isConnected = false;
String readLine = "";
String commandSplitter = "_NEXT_ERSCOMMAND_";
char commandSplitterChar[] = "_NEXT_ERSCOMMAND_";
bool testMessageSent = false;
EthernetClient client;

void setup()
{
}

void reconnect()
{
  isConnecting=true;
  Ethernet.begin(mac, ipLocal);
  Serial.begin(9600);
  delay(1000);
  serialPrint("Connecting...");

  if (client.connect(ipServer, port)) {
    client.flush();
    serialPrint("Connected.");
    isConnected = true;
  } else {
    serialPrint("Connection failed. Retry.");
  }
  isConnecting=false;
}

void loop()
{
  if (!isConnected) {
    if (!isConnecting) {
      reconnect();
    }
  } else {
    if (client.available()>0){
      clientReadIncomingMessage();
      serverWriteOutgoingMessage();
    }
  }
}
```

```

if (!isConnecting) {
    if (!client.connected()) {
        serialPrint("Disconnecting. Retry.");
        client.stop();
        isConnected = false;
    }
}
}

void clientReadIncomingMessage()
{
    char thisChar = client.read();
    readLine += String(thisChar);

    if (readLine.endsWith(commandSplitter)) {
        client.flush();
        serialPrint("Line received from server:");
        processIncomingMessage(readLine);
        readLine="";
    }
}

void processIncomingMessage(String im)
{
    String value = im;

    if (isDebugMode) {
        displayIncomingMessage(value);
    }

    // PROCESS YOUR INCOMING MESSAGES HERE
}

void displayIncomingMessage(String im)
{
    serialPrint(im);
}

void serverWriteOutgoingMessage()
{
    // SEND YOUR OUTGOING MESSAGES HERE

    if (!testMessageSent) {
        serialPrint("Sending test message...");
        client.write(prepareMessage("@Action.ServerTest.ers")); // WRITE AN @ACTION NAME TO THE SERVER TO EXECUTE IT
        testMessageSent=true;
    }
}

void serialPrint(String serialMessage)
{
    Serial.print(serialMessage);
    Serial.print('\n');
    Serial.flush();
}

char *prepareMessage(char result[]){
    strcat(result, commandSplitterChar);
    return result;
}

```

### 9.1.3. TCP serial connection (not recommended, please use API)

```
// Escaperoom Software Arduino demo project Serial
// Version 1.0 (8/22/2017)

// Sample compiled from various sources found on www.arduino.cc
// Modified by Escaperoom Software

// internal variables (don't change)
String readLine = "";
String commandSplitter = "_NEXT_ERSCOMMAND_";
bool testMessageSent = false;

void setup()
{
  reconnect();
}

void reconnect()
{
  Serial.begin(9600);
  while (!Serial) {}
}

void loop()
{
  //Send test message
  if (!testMessageSent) {
    serverWriteOutgoingMessage("@Action.ServerTest.ers"); // WRITE AN @ACTION NAME TO THE SERVER TO EXECUTE IT
    testMessageSent = true;
  }
  clientReadIncomingMessage();
  delay(25);
}

void serverWriteOutgoingMessage(String outgoingMessage)
{
  Serial.println(prepareMessage(outgoingMessage));
  Serial.flush();
}

String prepareMessage(String result){
  result = result + commandSplitter;
  return result;
}

void clientReadIncomingMessage()
{
  readLine += Serial.readString();
  Serial.flush();

  if (readLine.length()>0) {
    if (readLine.indexOf(commandSplitter)>0) {
      serverWriteOutgoingMessage("@Action.MicrocontrollerResponse.ers"); //Return an action to indicate the message
      was received by the Microcontroller
    }
    readLine = "";
  }
}
```

## 9.2. Raspberry PI

### 9.2.1. Send API call from Raspberry PI to ERS Server

The below will create a Python script that does an API call to ERS Server. Start ERS Server in Self-hosted API mode (see earlier chapter on this subject).

1. In Raspberry PI Console, install Requests Library  
Command: **pip install requests**

*Raspberry PI 4 and higher may require you to install components in a separate environment for security reasons.*

2. Write Python script sending an API call to ERS Server.  
Code example:

```
url = f'http://IP ADDRESS:8080/ersapi/runaction?roomID=&action=GUID'
response = requests.get(url)
if response.status_code == 200:
    print("Success:", response.text)
else:
    print("Error:", response.status_code)
```

### 9.2.2. Send API call from ERS Server to Raspberry PI

The below will create a Python script that runs a self-hosted API on the Raspberry PI which can then be called from ERS Server.

1. In Raspberry PI Console, install Flask.  
Command: **pip install Flask**

*Raspberry PI 4 and higher may require you to install components in a separate environment for security reasons.*

2. Write Python script running self-hosted API that can receive calls from external programs (such as the ERS Server). The API hosts a function called “ping” which will return the message “pong”.  
Code example:

```
from flask import Flask
app = Flask(__name__)
@app.route('/ping', methods=['GET'])
def ping():
    return "Pong!", 200
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080, debug=True)
```

### 9.2.3. Automatically start Raspberry PI software upon startup

You may want the Raspberry PI Python program to automatically when the Raspberry PI starts:

1. Create a new file using the the Console:

```
sudo nano /etc/systemd/system/flaskapp.service
```

2. Make the file similar to this:

```
[Unit]
```

```
Description=Flask App
```

```
After=network.target
```

```
[Service]
```

```
User=pi
```

```
WorkingDirectory=/home/pi/path/to/your/app >> actual app location
```

```
ExecStart=/usr/bin/python3 /home/pi/path/to/your/app/app.py >> actual app location
```

```
Restart=always
```

```
[Install]
```

```
WantedBy=multi-user.target
```

3. Now run these commands to enable the program as a startup service`

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable flaskapp.service
```

```
sudo systemctl start flaskapp.service
```

## 10. Frequently Asked Questions (FAQ)

### 10.1. How can I release my license key for migration to a new computer?

Please contact [support@escaperoombuilder.com](mailto:support@escaperoombuilder.com) and add your ordernumber and the license key from the original e-mail that you received on purchase.

### 10.2. Is there an Intercom/Camera feature

No. Both features are security intensive and network intensive. You should use a separate intercom/camera solution.

### 10.3. Suddenly my client shows a black screen, doesn't respond or behaves in a strange fashion

Please check the following:

- You may be running 2 or more rooms at the same time; please check whether the PORT number of the room (found in the settings) of each room is different, otherwise you will get "cross-communication" on the network. (please note that if you have a multi-computer set-up, the port number must be changed in the room settings of all devices)
- In very specific cases, a virus scanner may have blocked the PORT number. Try changing the port number to see if the room works correctly again (please note that if you have a multi-computer set-up, the port number must be changed in the room settings of all devices)

### 10.4. I want to have a 3<sup>rd</sup> computer+screen in my room that is connected to the Game Master PC (Server), but behaves differently

Here are some options on how to achieve this:

- True stand-alone
  - BASIC: A PC that just runs a video in a loop (not interactive)
  - ADVANCED: You may have separate software running on it that interacts with the Escaperoom Software API
- True integrated:
  - This requires 2 licenses
  - Create a separate room with it's own behaviour that runs on the same Port as the "Parent Room". In this way, the Game Master computer (Server) can run both different rooms simultaneously, but they can interact. Use Scripting-mode and Action GUIDs to trigger events between the "Parent Room" and "Child Room". If the action GUIDs are different, triggers will be broadcast to both rooms (because they are on the same Port), but since only 1 of the rooms has the action with the proper GUID, only that one will trigger. You may want to move all timer and message labels in the "Child room" below the bottom of the screen.