Hi! This tutorial shows you, the soon-to-be proud owner of a chatroom application, what the &^$*^*!” you are supposed to do to make one.

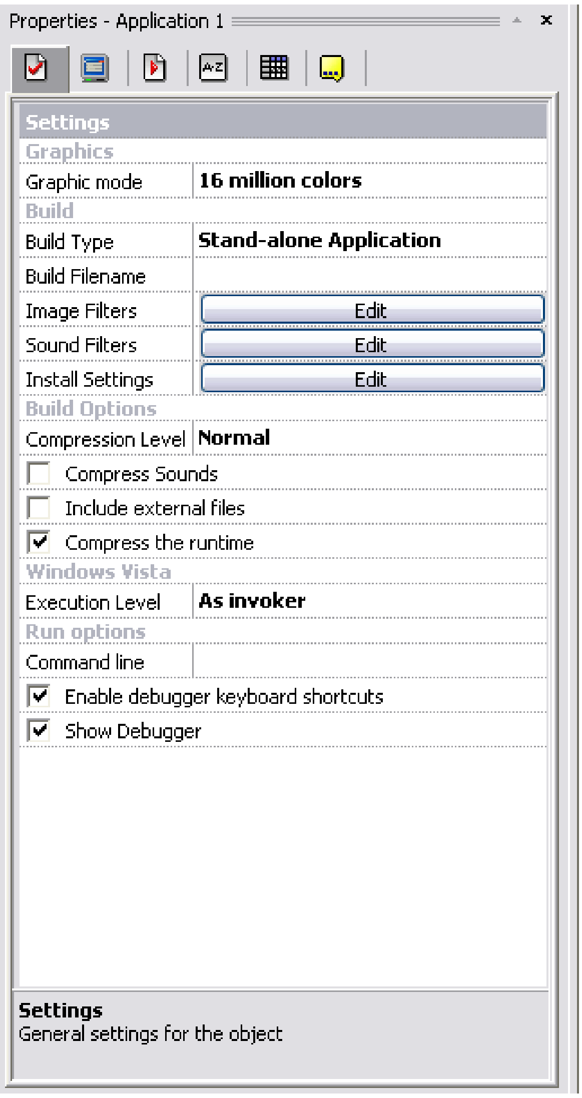
First – no, don't do that. Stop it. Really, leave it alone.

Good.

First, you need to create a new application, by pressing Ctrl+N.

Second, click the application in the Workspace Toolbar (Ctrl+W)Change the following properties of the application.

The Settings tab:



Properties - Application 1

Settings

Graphics

Graphic mode	16 million colors

Build

Build Type	Stand-alone Application
Build Filename	
Image Filters	Edit
Sound Filters	Edit
Install Settings	Edit

Build Options

Compression Level	Normal

☐ Compress Sounds
☐ Include external files
☑ Compress the runtime

Windows Vista

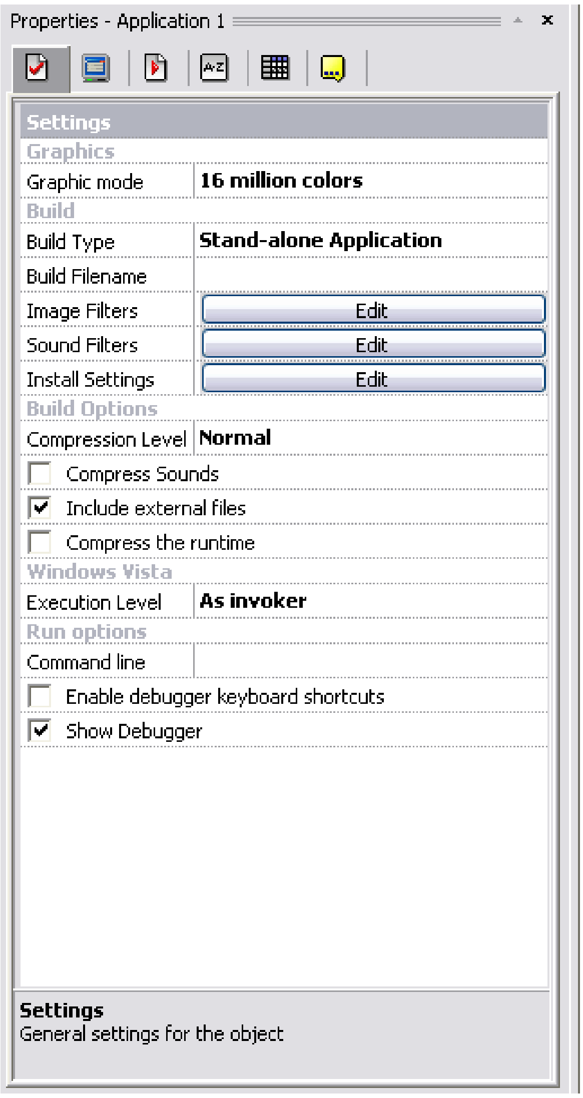
Execution Level	As invoker

Run options

Command line	

☑ Enable debugger keyboard shortcuts
☑ Show Debugger

Settings
General settings for the object



Properties - Application 1

Settings

Graphics

Graphic mode	16 million colors

Build

Build Type	Stand-alone Application
Build Filename	
Image Filters	Edit
Sound Filters	Edit
Install Settings	Edit

Build Options

Compression Level	Normal

☐ Compress Sounds
☑ Include external files
☐ Compress the runtime

Windows Vista

Execution Level	As invoker

Run options

Command line	

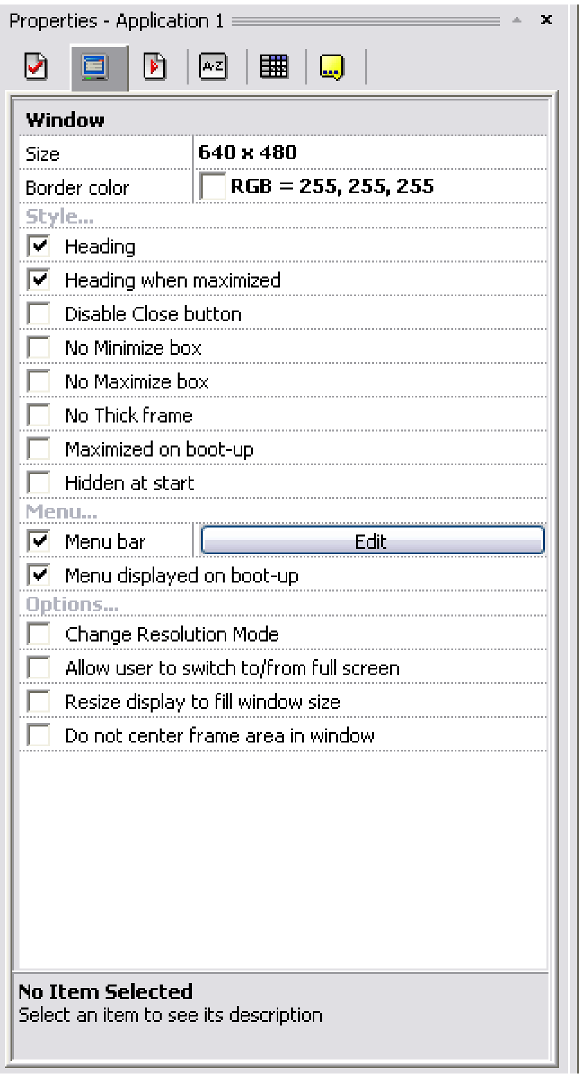
☐ Enable debugger keyboard shortcuts
☑ Show Debugger

Settings
General settings for the object

- Include external files: **Check**. This will make sure that all files you add later (for example, On Message sound) will be included in the installer automatically.
- Compress the runtime: **Uncheck**. This is a essential thing to do with any Internet-accessing application; the main program usually extracts to a temporary (randomly generated) folder, making it use up less memory. If you have a firewall enabled, this will set it off every time it runs, as the temporary folder has a random name – so every time it will change, and your firewall will detect this as a new app trying to connect.
- Enable debugger keyboard shortcuts: **Uncheck**. This is a personal preference, and recommended if you are a beginner; unchecking this will stop any accidental key combination pressing from doing anything horrible with your app's debugger

Switch to the second tab, Window.

The Window Tab:



Properties - Application 1

Window

Size: **640 x 480**

Border color: □ **RGB = 255, 255, 255**

Style...

☑ Heading
☑ Heading when maximized
□ Disable Close button
□ No Minimize box
□ No Maximize box
□ No Thick frame
□ Maximized on boot-up
□ Hidden at start

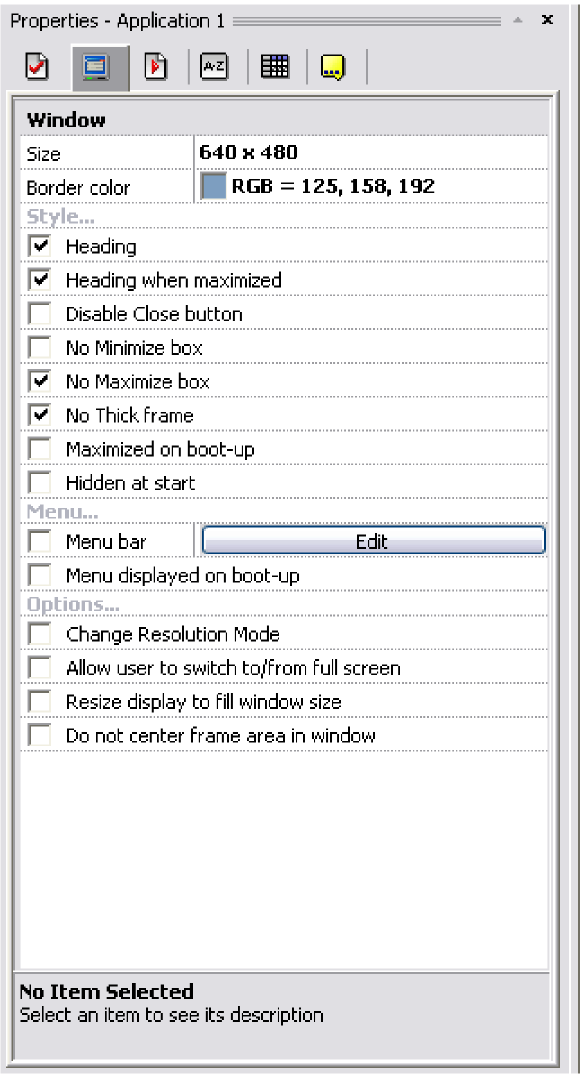
Menu...

☑ Menu bar [Edit]
☑ Menu displayed on boot-up

Options...

□ Change Resolution Mode
□ Allow user to switch to/from full screen
□ Resize display to fill window size
□ Do not center frame area in window

No Item Selected
Select an item to see its description



Properties - Application 1

Window

Size: **640 x 480**

Border color: ■ **RGB = 125, 158, 192**

Style...

☑ Heading
☑ Heading when maximized
□ Disable Close button
□ No Minimize box
☑ No Maximize box
☑ No Thick frame
□ Maximized on boot-up
□ Hidden at start

Menu...

□ Menu bar [Edit]
□ Menu displayed on boot-up

Options...

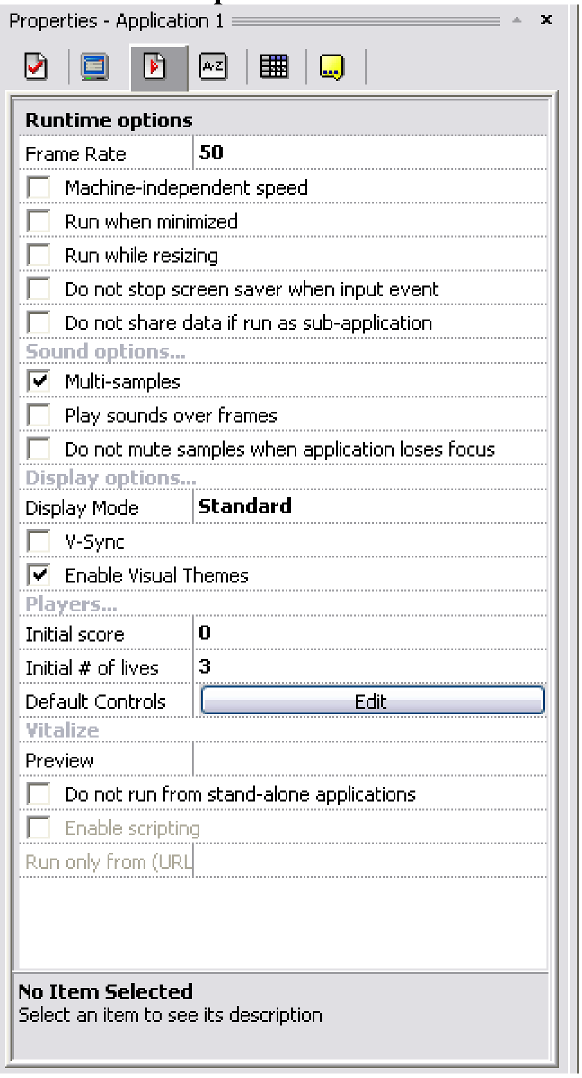
□ Change Resolution Mode
□ Allow user to switch to/from full screen
□ Resize display to fill window size
□ Do not center frame area in window

No Item Selected
Select an item to see its description

- Border colour: **Change to anything non-basic.** Using a basic colour will make your app unpopular – Bright/Full/Basic colours are an absolute no-no. You may, however, use black – for some strange reason everyone loves it.
- No maximize box: **Check.** If people maximize your app, they will see the frame surrounded by the border colour. Not very professional. Checking this means the user cannot maximize your app.
- No thick frame: **Check.** Again, if people change the size of your app, they will see the frame surrounded by the border colour. What's worse, they can resize the window to be too small, meaning that they can't see some of the frame, and its very hard to readjust it to what it used to be. Checking this means the user cannot resize your app's window.
- Menu bar: **Uncheck.** This is not used in the tutorial. Uncheck to avoid being unprofessional. This will hide the application's menu, usually located just below the window's title.
- Menu displayed on boot-up: **Uncheck.** This is not used in the tutorial. Uncheck to avoid being unprofessional. This will stop the menu (last bullet point) from appearing on start-up of the app.

Move onto the next tab, Runtime options.

The Runtime Options Tab:



Properties - Application 1

Runtime options

Frame Rate **50**

□ Machine-independent speed

□ Run when minimized

□ Run while resizing

□ Do not stop screen saver when input event

□ Do not share data if run as sub-application

Sound options...

☑ Multi-samples

□ Play sounds over frames

□ Do not mute samples when application loses focus

Display options...

Display Mode **Standard**

□ V-Sync

☑ Enable Visual Themes

Players...

Initial score **0**

Initial # of lives **3**

Default Controls Edit

Vitalize

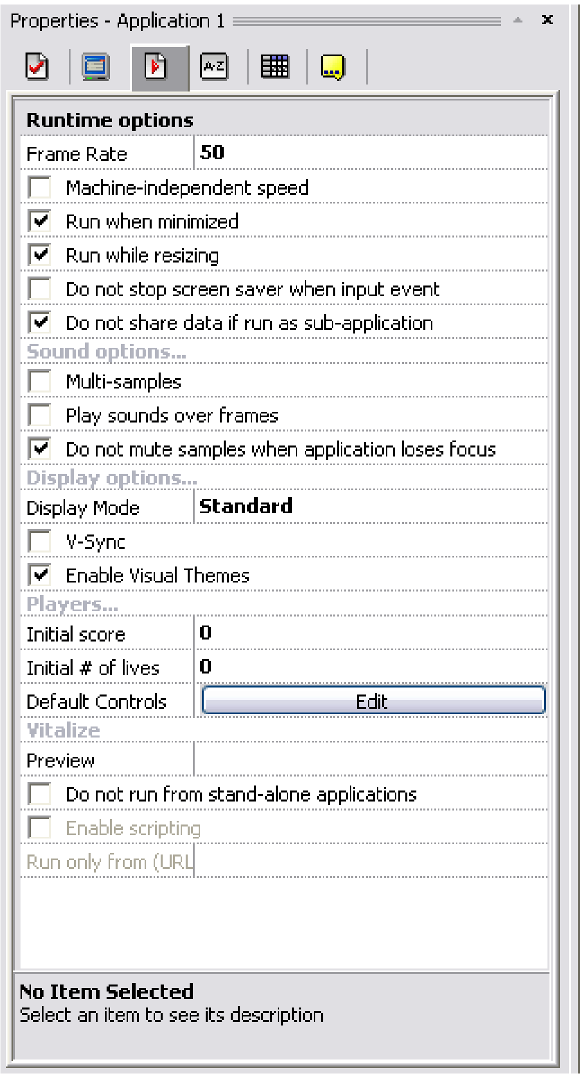
Preview

□ Do not run from stand-alone applications

□ Enable scripting

Run only from (URL

No Item Selected
Select an item to see its description



Properties - Application 1

Runtime options

Frame Rate **50**

□ Machine-independent speed

☑ Run when minimized

☑ Run while resizing

□ Do not stop screen saver when input event

☑ Do not share data if run as sub-application

Sound options...

□ Multi-samples

□ Play sounds over frames

☑ Do not mute samples when application loses focus

Display options...

Display Mode **Standard**

□ V-Sync

☑ Enable Visual Themes

Players...

Initial score **0**

Initial # of lives **0**

Default Controls Edit

Vitalize

Preview

□ Do not run from stand-alone applications

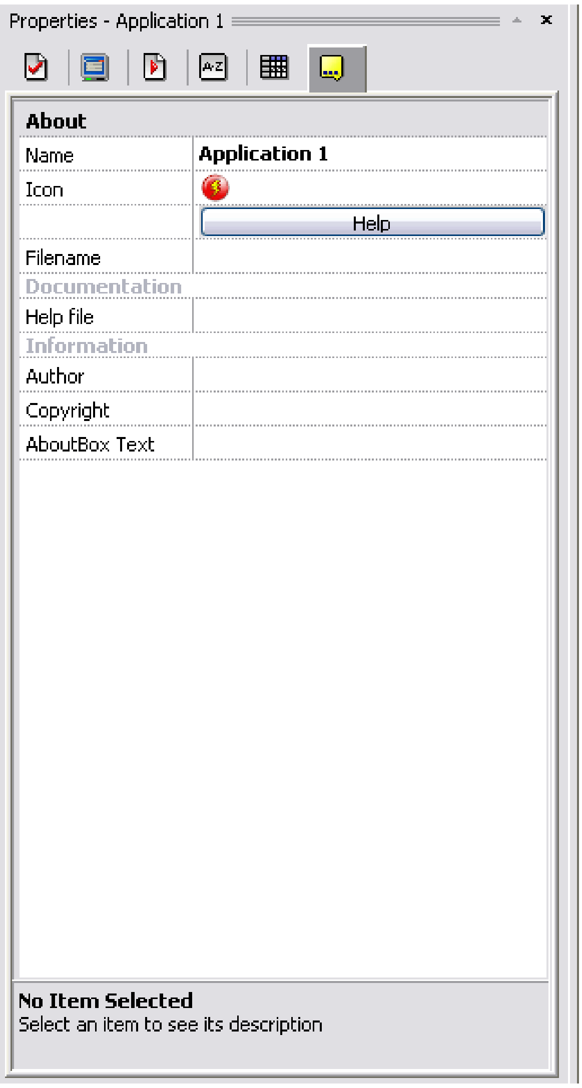
□ Enable scripting

Run only from (URL

No Item Selected
Select an item to see its description

- Run when minimized: **Check.** An essential setting to use, as this lets the application run when its minimized...if it doesn't run, the program may receive no messages and/or terminate the connection to the server. Either way, not good.
- Run when resizing: **Check.** This one doesn't just apply to resizing the app; it can also apply to changing data in memory – for example, holding down a button. If it doesn't run, the program may receive no messages and/or terminate the connection to the server.
- Do not share data if run as a sub-application: **Check.** This is a weak blocker for hacking attempts, it stops accessing of data in certain parts of the program, by certain other programs. If you are running this program using the Sub-Application object, then you should leave it unchecked. Otherwise, check it.
- Multi-samples: **Uncheck.** This is unneeded unless you plan to play more than one sound at a time – which will usually make all of the sounds inaudible.
- Do not mute samples when application loses focus: **Check.** This ensures that if the app plays a music file (certain types are known as samples) and the application is minimized or not selected by the user, it will still play, rather than automatically stopping.
- Initial # of lives: Set to 0. This changes the number of lives to 0 – a personal preference.
- Default Controls: Set by clicking the Edit button, then the X in each column, and then changing each key to 1. (1 is a random key.) This is also a personal preference.

The About Tab:



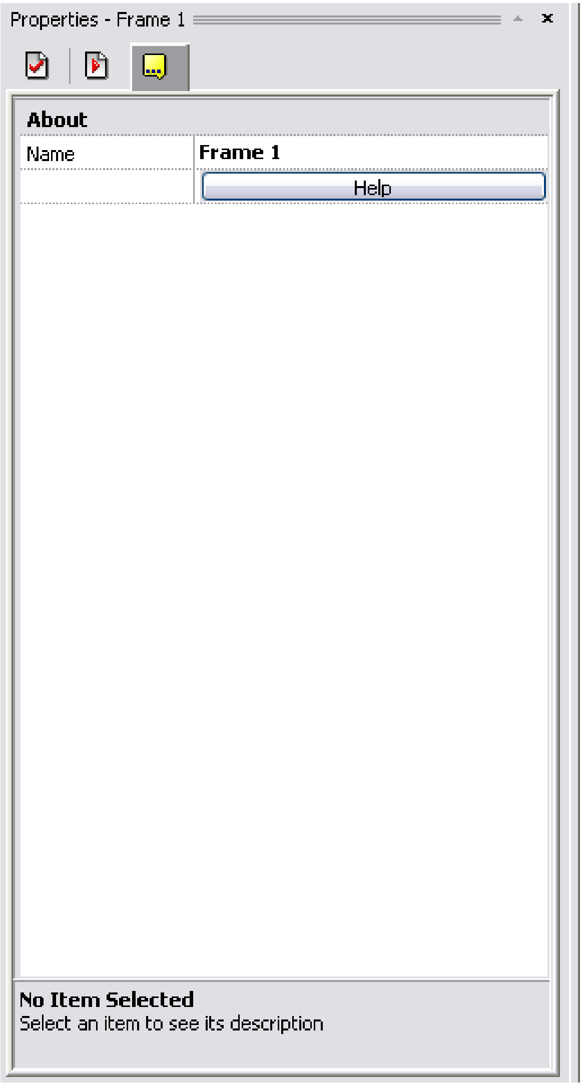
Properties - Application 1

About	
Name	**Application 1**
Icon	
	Help
Filename	
Documentation	
Help file	
Information	
Author	
Copyright	
AboutBox Text	

No Item Selected
Select an item to see its description

- Name: Change to what you want your application name to be. I picked The Cool Chat Program, but that's just me
- Icon: Change to what you want your application icon to be. I just copied the icons into Paint and inverted the colours (Ctrl+I).
- Filename: To change this, save the MFA by pressing Ctrl+S.
  (Note about Help file: Although there is a help file, there's no point putting it in here – as it will most likely be installed to somewhere else.)
- Author: Enter your nickname and/or your first and last name.
- Copyright: put ©[year] by [Author box text] - "©2010 by SortaCore" for me.
  (To insert a ©, hold down left Alt, and press 0169 on the Number Pad, then release Alt.)

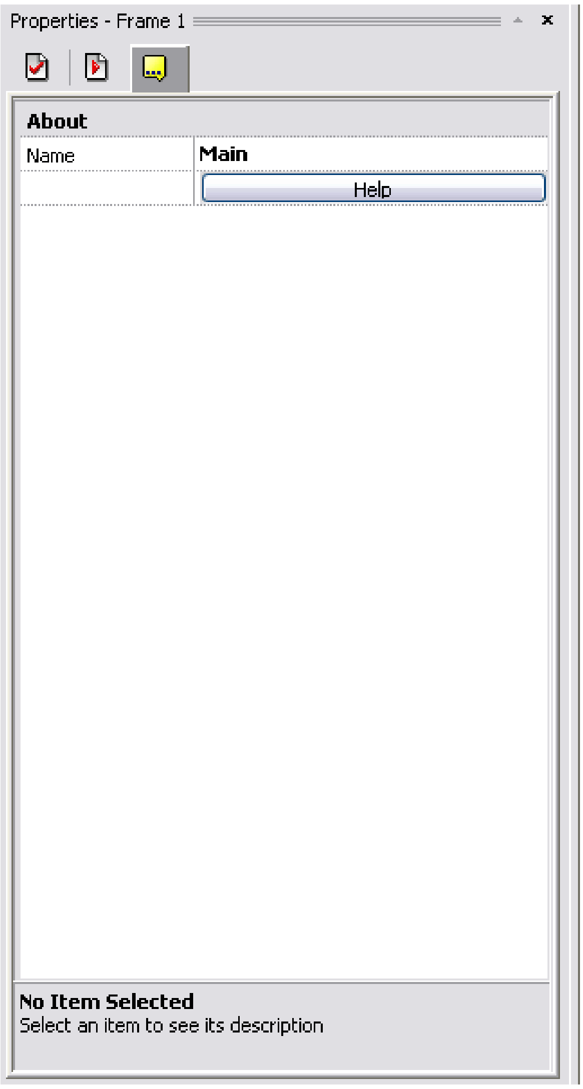
Click the Frame and select the third tab. Change the name (Main is my default when the application has only one frame).



About

Name	**Frame 1**
	Help

No Item Selected
Select an item to see its description

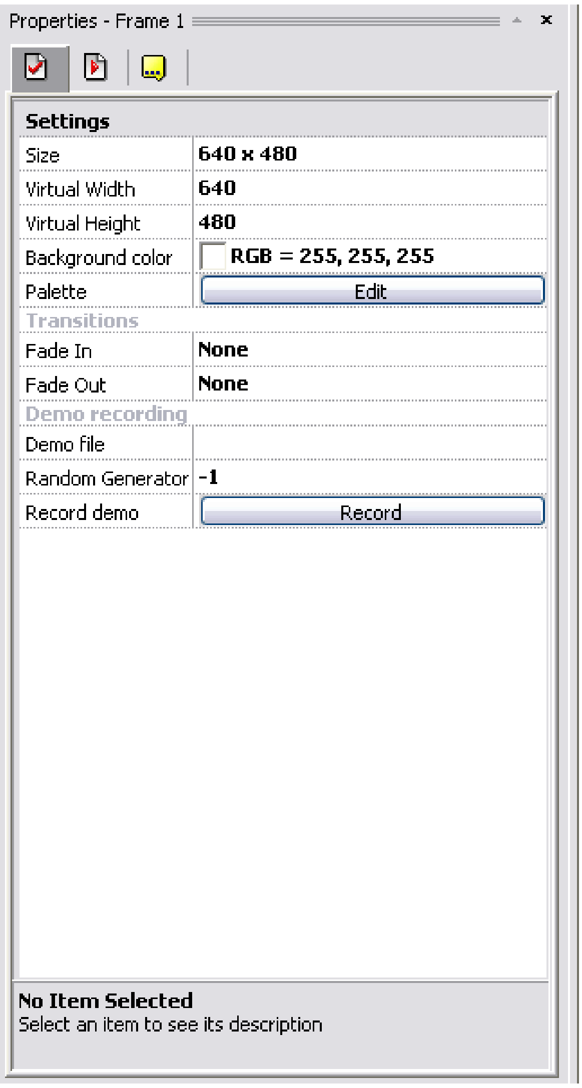


About

Name	**Main**
	Help

No Item Selected
Select an item to see its description

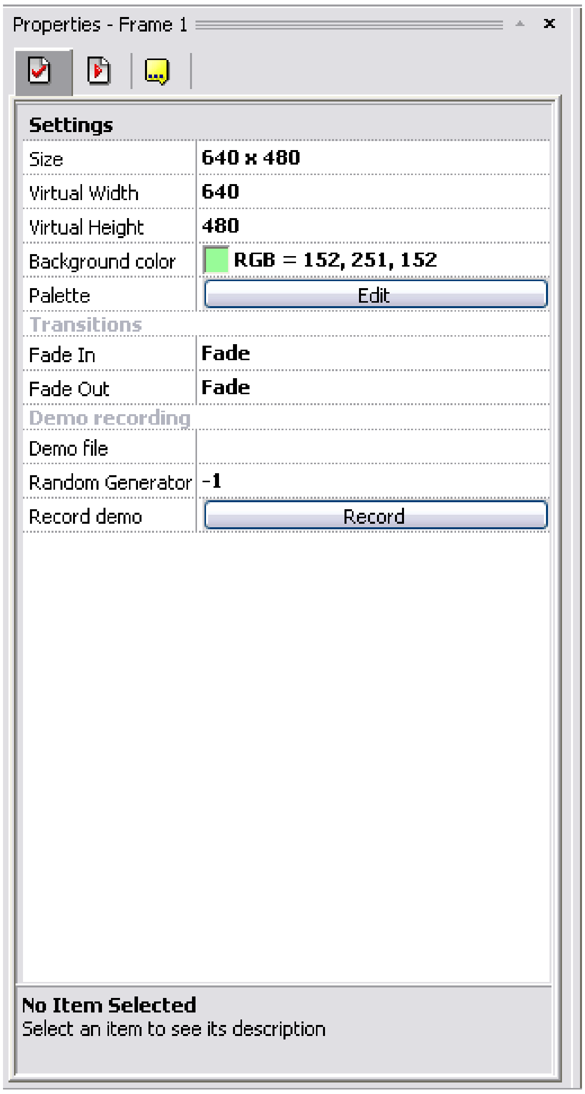
The Settings Tab:



Properties - Frame 1

Settings	
Size	640 x 480
Virtual Width	640
Virtual Height	480
Background color	□ RGB = 255, 255, 255
Palette	Edit
Transitions	
Fade In	None
Fade Out	None
Demo recording	
Demo file	
Random Generator	-1
Record demo	Record

No Item Selected
Select an item to see its description



Properties - Frame 1

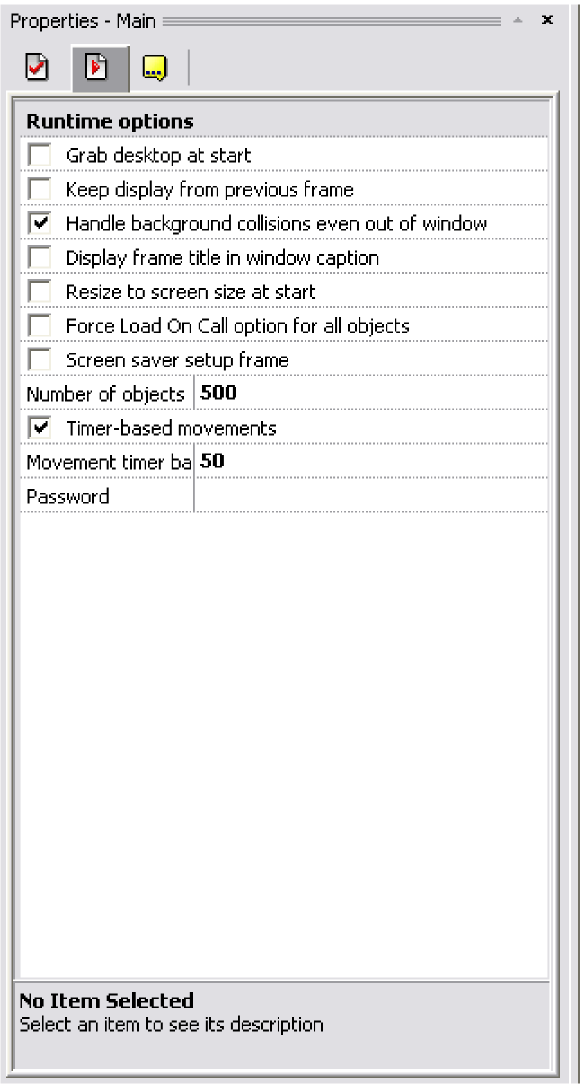
Settings	
Size	640 x 480
Virtual Width	640
Virtual Height	480
Background color	🟩 RGB = 152, 251, 152
Palette	Edit
Transitions	
Fade In	Fade
Fade Out	Fade
Demo recording	
Demo file	
Random Generator	-1
Record demo	Record

No Item Selected
Select an item to see its description

- Background colour: **Change to any non-basic colour**; one that's not basic, bright or full, excluding the cool black.
- Fade In: **Change to any setting.** This changes the way the frame starts up in the application. My default one is fade from black, 1 second.
- Fade Out: **Change to any setting.** This changes the way the frame ends in the application. My default one is fade to black, 1 second.

Switch to the second tab, Runtime options.

The Runtime Options Tab:



Properties - Main

Runtime options

☐ Grab desktop at start

☐ Keep display from previous frame

☑ Handle background collisions even out of window

☐ Display frame title in window caption

☐ Resize to screen size at start

☐ Force Load On Call option for all objects

☐ Screen saver setup frame

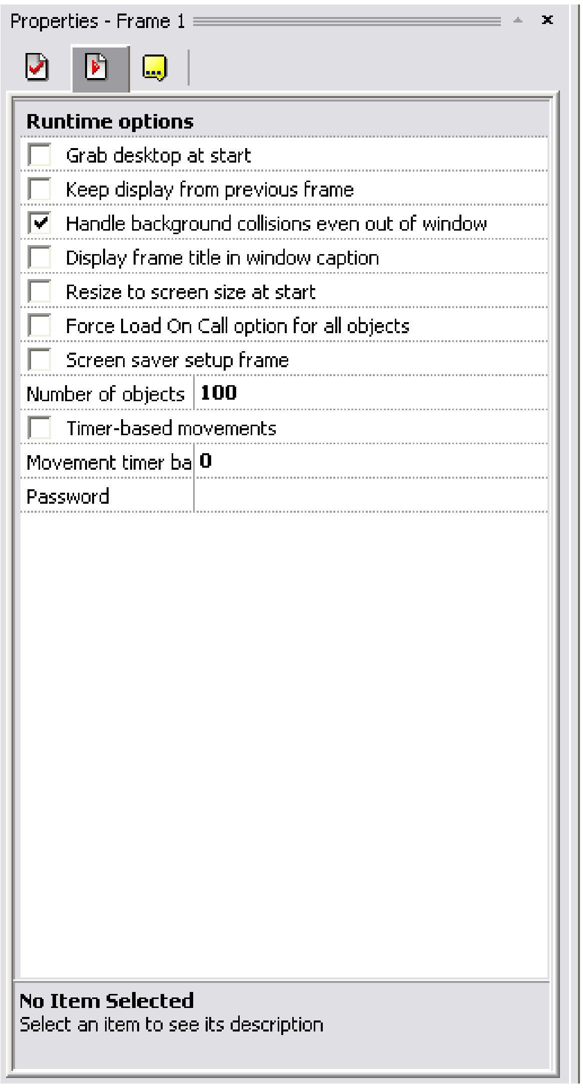
Number of objects **500**

☑ Timer-based movements

Movement timer ba **50**

Password

No Item Selected
Select an item to see its description



Properties - Frame 1

Runtime options

☐ Grab desktop at start

☐ Keep display from previous frame

☑ Handle background collisions even out of window

☐ Display frame title in window caption

☐ Resize to screen size at start

☐ Force Load On Call option for all objects

☐ Screen saver setup frame

Number of objects **100**

☐ Timer-based movements

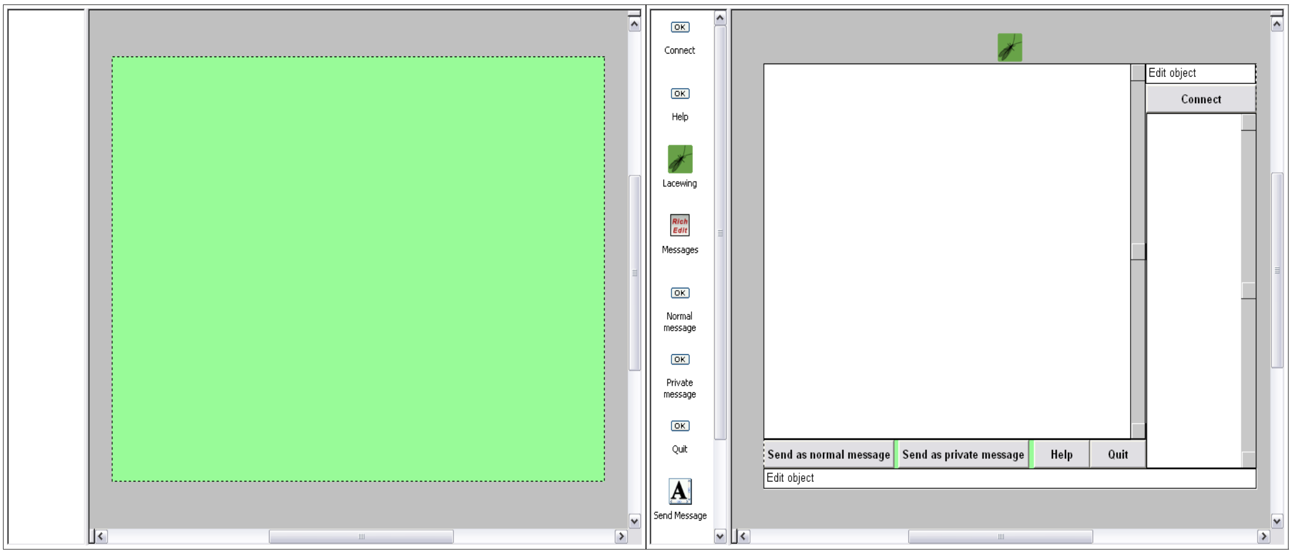
Movement timer ba **0**

Password

No Item Selected
Select an item to see its description

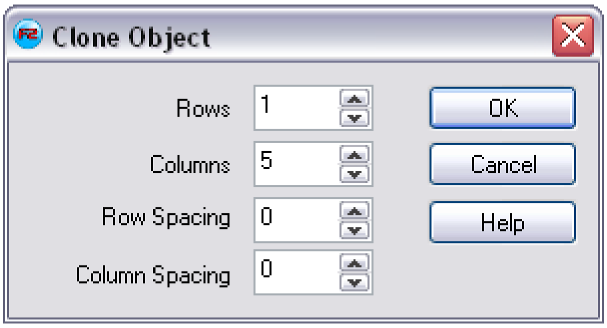
- Number of objects: **Change to 100.** This is the maximum number of objects the memory will use – as we will be using about 10, we change it to the minimum, 100. This has a small effect on the memory usage – professional.
- Timer-based movements: **Uncheck.** This will disable the timer for movements – there is no movements in this tutorial.
- Movement timer based: **Set to 0.** This is mainly a personal preference, to make it easier to confirm the movement timer is disabled.

Now the good stuff: Double-click the frame in the Workspace window, and then press Ctrl+W (hide the workspace window – its not needed any more).



The Frame Editor Configuration

Wanna know how to make that difference?



Rows: 1
Columns: 5
Row Spacing: 0
Column Spacing: 0

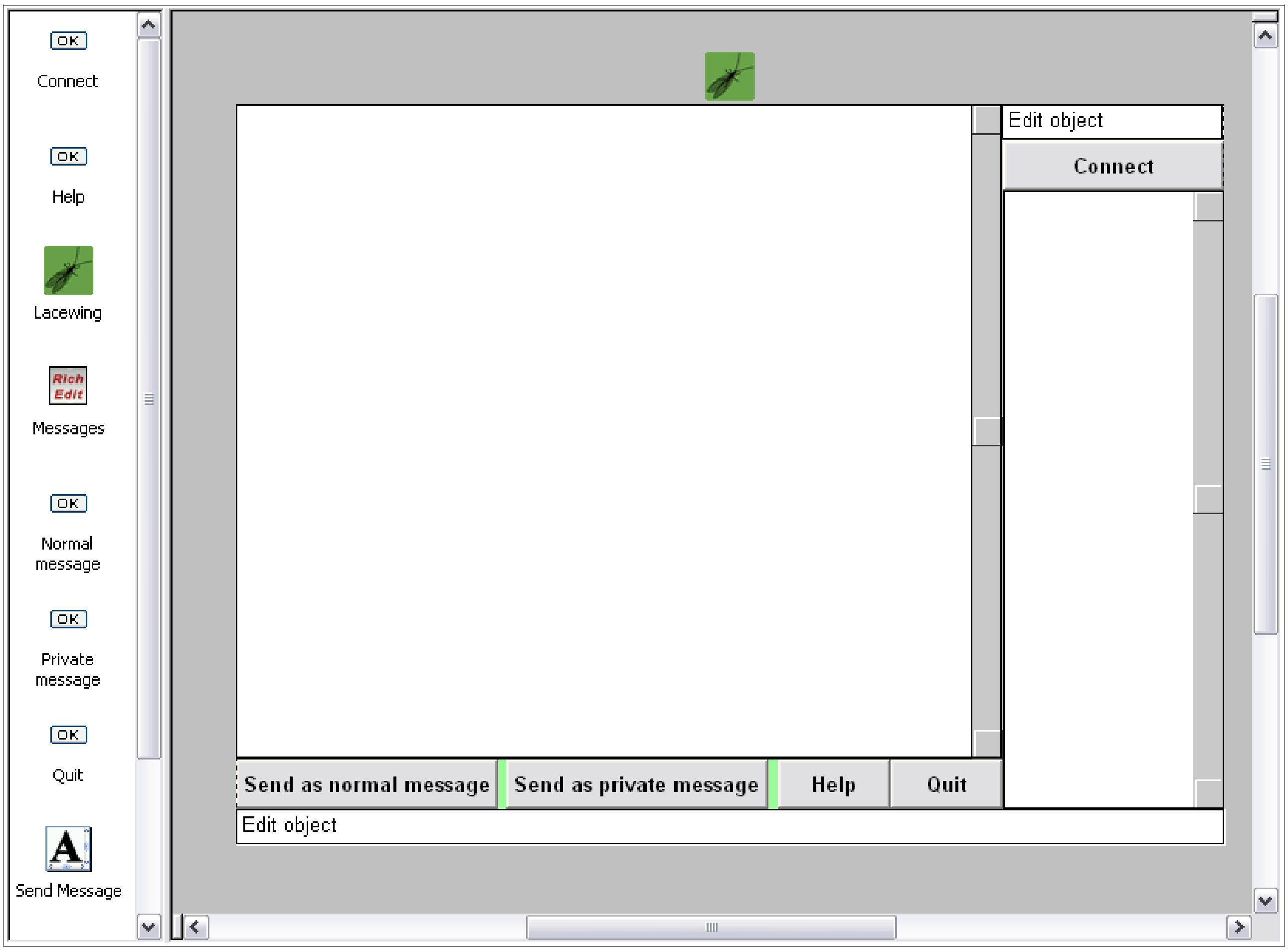
Follow these steps:
Step 1: Create a button object.
Step 2: Make the button text bold (3rd tab).
Step 3: Then clone it to these settings: → →
Step 4: Create a Rich Edit object.
Step 5: Switch to first tab on the Rich Edit object, then change the following settings:

- Use System Colours: **Uncheck.** This will allow the Rich Edit object to use any colour it wants.
- No colour change when disabled: **Check.** This allows the Rich Edit object to keep its
- colour when its window is disabled.
- Read only: **Check.** This will disallow the user to edit or overwrite the text.
- Use features: **Check.** This will allow the option below.
- Automatic detection of URLs: **Check.** This will make URLs change to their default font setting: blue and underlined.

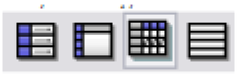
Step 6: Create an Editbox object.
Step 7: Switch to first tab on the Editbox object, then **check** the Read-only setting.
Step 8: Clone the object (2 rows, 1 column).
Step 9: Create an Lacewing object (downloadable from http://lacewing.aquadasoft.com/Lacewing.zip in case you don't have it already)
Step 10: Create a list object
Step 11: Renaming (switch to last tab):

- Button 1 → Connect.
- Button 2 → Normal Message.
- Button 3 → Private Message.
- Button 4 → Help
- Button 5 → Quit
- Rich Edit object → Messages
- Editbox 1 → Send Message
- Editbox 2 → Username
- Lacewing client → Lacewing
- List → Users online

Step 12: Positioning and resizing:
Connect: (497,24); 143 by 32
Normal Message: (0,425); 170 by 32
Private Message: (175,425); 170 by 32
Help: (351,425) 73 by 32
Quit: (424,425); 73 by 32
Messages: (0,0); 494 by 424
Message to send: (0,457); 640 by 23
Username: (496,0); 143 by 23
Users online: (497,56); 142 by 400
Step 13: Select Lacewing and switch to the first tab. **Check** Automatically clear stack – this means that all the messages don't wait to be sent and/or are kept in memory after being sent.
You have successfully created the chatroom's frame layout!
The result should look like this:

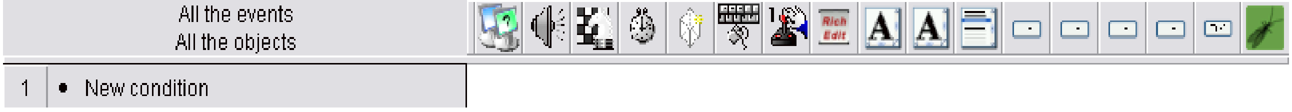


Now we move onto the gears and levers of the chatroom – the Events, editable in the Event Editor. Switch to the Event Editor by clicking this button:



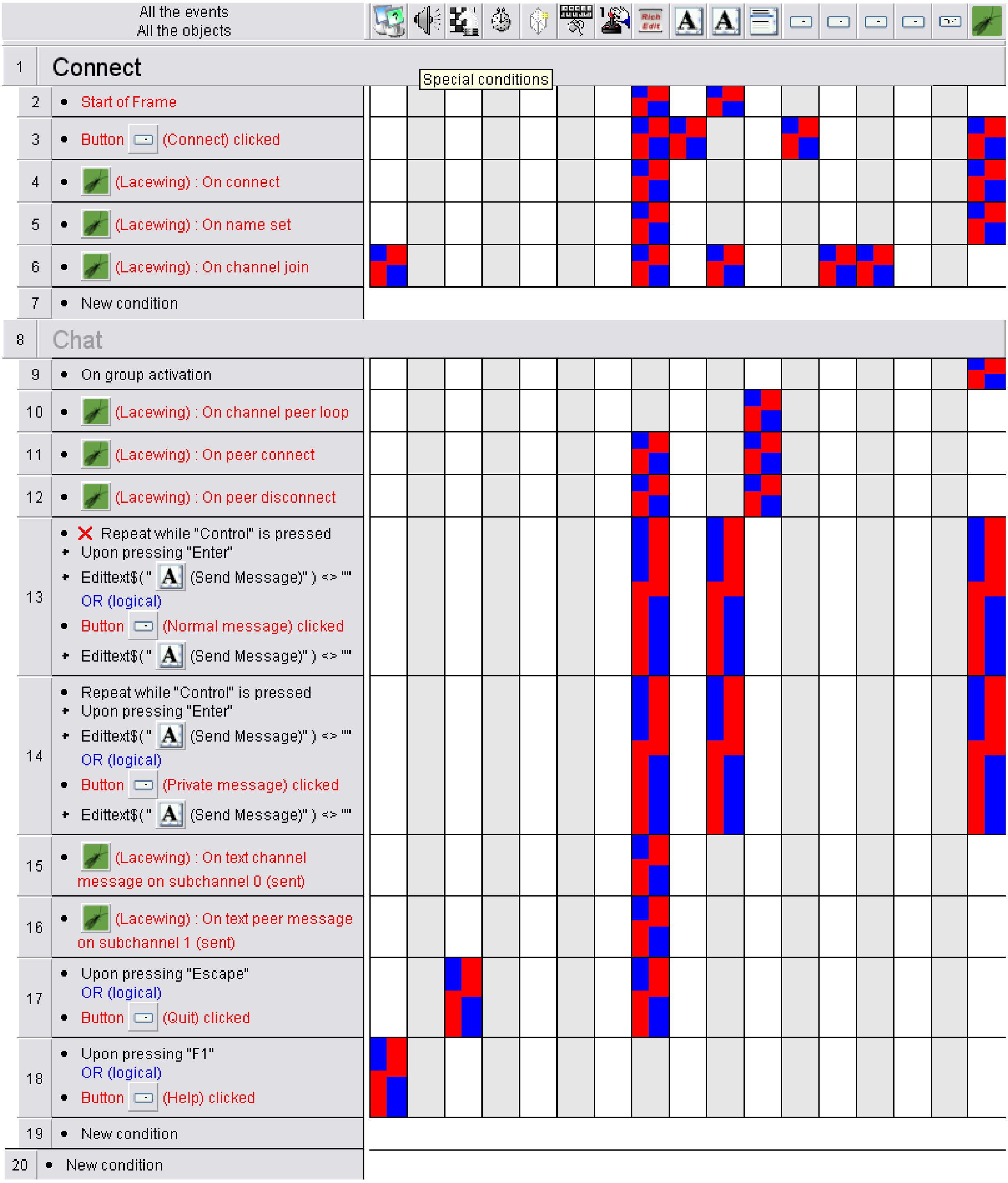
↖ This one

The Event Editor Configuration



All the events
All the objects

1 • New condition



All the events
All the objects

1 Connect

2 • Start of Frame

3 • Button (Connect) clicked

4 • (Lacewing) : On connect

5 • (Lacewing) : On name set

6 • (Lacewing) : On channel join

7 • New condition

8 Chat

9 • On group activation

10 • (Lacewing) : On channel peer loop

11 • (Lacewing) : On peer connect

12 • (Lacewing) : On peer disconnect

13
• ✕ Repeat while "Control" is pressed
• Upon pressing "Enter"
• Edittext$(" A (Send Message)") <> ""
OR (logical)
• Button (Normal message) clicked
• Edittext$(" A (Send Message)") <> ""

14
• Repeat while "Control" is pressed
• Upon pressing "Enter"
• Edittext$(" A (Send Message)") <> ""
OR (logical)
• Button (Private message) clicked
• Edittext$(" A (Send Message)") <> ""

15 • (Lacewing) : On text channel message on subchannel 0 (sent)

16 • (Lacewing) : On text peer message on subchannel 1 (sent)

17
• Upon pressing "Escape"
OR (logical)
• Button (Quit) clicked

18
• Upon pressing "F1"
OR (logical)
• Button (Help) clicked

19 • New condition

20 • New condition

If you are a careful reader, or just very boring, you'll notice there's a lot. Let's walk and talk.
This is the format the events will be written in:
Objects will be written in blue, unless they are a parameter.
String parameters will be written in red.
Value parameters will be written in brown.
Option parameters will be written in green.

Event #
+("Object") Condition 1
+("Object to retrieve condition from") Condition 2
-Object: Action 1
-Object 2: Action 2
That is an example. Do not hold it to me, even if you want to. Let's get on with it.
Create 2 groups of events (GoE) and set one:
Name: Connect.
Activate on frame start: Yes.
And the second:
Name: Chat.
Activate on frame start: No.

Event #2 (Event #1 is the Connect GoE.)
+Frame object: Start of Frame
-Send Message: Control>Read only> On. *This stops fools trying to send messages before they're connected.*
-Messages: Control>Set Text to Loading... *This is because all other Set Text will begin with "Newline$+". This makes the Messages object look more professional as it doesn't have a empty line at the top of it.*

Event #3
+Connect: Button clicked
-Connect: Disable.
-Lacewing>Connect to server aquadasoft.com on port 6121. *This is a well-used public server.*
-Username: Control>Read only>On. *This stops fools trying to change their username while the program's connecting, then getting bothered when they're halfway through editing it when it runs the Set Name command and they're stuffed with half the username that they wanted.*
-Messages: Control>Set Text to Newline$+Connecting... *If we don't keep everyone up-to-date on what's happening, it looks like its doing nothing...not good.*

Event #4
+Lacewing: Connection>On Connect
-Messages: Control>Set Text to Newline$+Setting name... *Keep 'em up to date*
-Lacewing: Set Name to Edittext$("Username") *This will also work for renaming a user, but we're not using renaming in this chatroom.*

Event #5
+Lacewing: Channel>On join
-Send Message: Control>Read only>Off. *This allows everyone to put characters into the editbox so they can chat.*
-Normal message: Enable. *This allow the Send Normal Message button to work.*
-Private message: Enable. *This allow the Send Private Message button to work.*
-Messages: Control>Set Text to Newline$+Channel joined, you may now chat. *Keep 'em up-to-date*
-Special object: Group of events: Deactivate Connect *Otherwise something might go wrong*
-Special object: Group of events: Activate Chat *Now the Send Message commands will work*
Now the next GoE: Chat.

Event #9
+Special object:Group of events>On group activation
-Lacewing: Peers>Select>Loop *This will allow us to list all the users on the channel and add them to the Users online object using event #10.*

Event #10
+Lacewing>Channel>Peer>On peer loop
-Users online: Add line Peer_Name$(“Lacewing”)

Event #11
+Lacewing>Channel>Peer>On peer connect
-Users online: Add line Peer_Name$(“Lacewing”) *This adds the joined user to the list, informing the program's owner that someone joined (next action)*
-Messages:Control>Set Text Newline$+“**** ”+Peer_Name$(“Lacewing”)+“ Enters Chat ****”

Event #12
+Lacewing>Channel>Peer>On peer disconnect
-Users online: Delete line FindStringExact(“Users online”, Peer_Name$(“Lacewing”), -1) *This deletes the user that left from the list, informing the program's owner that someone left (next action) By the way, the -1 indicates that the search for the string should start at the start of the list object.*
-Messages: Control>Set Text Newline$+“**** ”+Peer_Name$(“Lacewing”)+“ Left Chat ****”

Event #13
+Mouse & Keyboard>Keyboard>Repeat while Control is pressed [Negate] *This is the shortcut key*
+Mouse & Keyboard>Keyboard>Upon pressing Enter *for normal messages.*
+Special object: Compare two general values>Edittext$(“Send Message”) <> “”
OR (logical) *<> means Different to.*
+Normal message: Button clicked *This is the other way of triggering the Send Message command.*
+Special object: Compare two general values>Edittext$(“Send Message”) <> “”
-Lacewing: Send>Text>To channel: Edittext$(“Send Message”) on subchannel 0
-Messages: Control>Set Text to Newline$+“<”+Self_Name$(“Lacewing”)+”> ”+Edittext$(“Send Message”) *<Messaging User> Example message format.*
-Send Message: Control>Set Text “” *This stops the user from having to clear the Send Message box themselves – more professional.*

Event #14
+Mouse & Keyboard>Keyboard>Repeat while Control is pressed This is the shortcut key for
+Mouse & Keyboard>Keyboard>Upon pressing Enter private messages (Ctrl+Enter)
+Special object: Compare two general values>Edittext$(“Send Message”) <> “”
OR (logical)
+Private message: Button clicked
+Special object: Compare two general values>Edittext$(“Send Message”) <> “”
-Lacewing: Peer>Select>By name: List Select$(“Users online”) *The current selected user on the list.*
-Lacewing: Send>Text>To peer: Edittext$(“Send Message”) on subchannel 1 *1 just because.*
-Messages: Control>Set Text to Newline$+“<”+List Select$(“Users online”)+“: ”+Edittext$(“Send Message”)
-Send Message: Control>Set Text “”

Event #15
+Lacewing: Message>Sent>On text channel message, subchannel 0
-Messages: Control>Set Text to Newline$+“<”+Peer_Name$(“Lacewing”)+“> ”+Received$ (“Lacewing”) *This means that you will receive normal messages in much the same way you send them.*

Event #16
+Lacewing: Message>Sent>On text peer message, subchannel 1

-Messages: Control>Set Text to Newline$+“>”+Peer_Name$(“Lacewing”)+“: ”+Received$ (“Lacewing”) *Private messages are displayed differently when received than when sent. This is because, when sending, you need to know who you're sending it to, and what you're sending; when you're receiving, however, you need to know who sent it and what the message was.*

Event #17

+Mouse & Keyboard>Keyboard>Upon pressing Escape *This is the Quit event.*
OR (logical) *Important.*
+Quit: Button clicked
-Messages: Files>Save RTF Text Apppath$+“Log.rtf”
-Frame object: End the application

Event #18

+Mouse & Keyboard>Keyboard>Upon pressing F1 *This will bring up MMF2 help, just close it again. This is the Help event, important too.*
OR (logical)
+Help: On button clicked
-Special object: Execute an external program>With an evaluated pathname:
Program name: notepad.exe
Command-line: Apppath$+“help.txt”
Hide and/or wait: Hide.

Well done! Your application now works, and is a efficient chatroom, although basic.
There are still several optional features you can add. The most important one to think of is the help file. To create it, press WinKey+R, type notepad.exe and then edit the text document. Make sure you save it in your application's directory, as help.txt!

Author's notes:
Creator: SortaCore
Program: Basic chatroom
Features: Normal chatting, private chatting, connecting and disconnecting, using Lacewing protocol.
Frames: 1.
Frame 1 Name: Main.
Frame 1 objects: 10.
Frame 1 object species: 5.
GoE count: 2.
Event count: Exclusive 15, inclusive 20.
Author's comments:
“I am the Manager and Head Programmer for Darkwire Software. Please visit my site at http://tinyurl.com/darkwire”