

# Create It With Live **CODE**

*in cooperation with*



DIGITAL  
POMEGRANATE

# Hello World



## Day 3 : Hello Controls

## Say Hello to the...Tools Palette in Depth

in the last two days, you have been adding controls to Cards to create our Hello World App. This was done using the tools palette that allows you to change between Edit (top right) and Run (top left) mode, add controls, and edit bitmap images with the paint tools.

To get recap, you drag any control onto a card. You can also double-click an object to create a object of that type in the center of the card. But first lets do a review of the available controls





LIVE **CODE**

Hello World App

## Control Resources

[Devin Asay](#), from Brigham Young University, has done a wonderful job creating [Learning to Program with Livecode](#). This is based on his college course and you will find a great deal of well written explanations and suggestions on coding with LiveCode.



[LiveCode Objects](#)

[Properties of Objects in LiveCode](#)

[Scripting Object Properties](#)

[Groups](#)

# Buttons

Buttons are fundamental entities that allow you to interact with devices, for example with mechanical buttons on a keyboard or a remote control, or graphical buttons on a touch sensitive computer display

All button objects in LiveCode are highly flexible and customizable. Common settings include the ability to show and hide the border or fill, and to display an icon.



-  Push button
-  Check box button
-  Radio button
-  Default button
-  Rectangle button
-  Tab Panel button



# Fields

There are 5 basic field types in LiveCode, all of which you can add to your stack from the Tools Palette.

**Label field:** a single line, non-editable field

**Text entry field:** a single line, editable field for user data entry.

**Scrolling field:** a multi line, scrollable, editable field.

**Basic table field:** a basic table field allowing data to be displayed in cells

**Scrolling list field:** a multi line, scrollable, non-editable field allowing the user to make single or multiple selections



Label field



Text entry field



Scrolling field



Basic table field



Scrolling list field

# Menus

Menus are used to display a list of choices. The pulldown menu displays a standard pulldown menu, The option menu allows a choice from a list. The combobox allows the user to type an option or choose from a list. Popup menus can be displayed underneath the cursor and used to provide context sensitive options anywhere in your application.



-  Option Menu
-  Pulldown Menu
-  Combo box
-  Pop Up Menu

# Scrollbars

Scroll bars can be used as a *progress bar* to display a value, a *slider* to allow the user to choose a value, or to scroll objects. (Note that you don't need to use a scrollbar object with fields or groups as these can display a built-in scroll bar.) Sliders and scrollbars can be displayed both horizontally and vertically – to display vertically, resize so that the height is greater than the width.



 Progress Bar

 Slider

 Little Arrows

 Scrollbar



## Hello World App

# Images

LiveCode supports a wide range of media formats, allowing you to produce rich media applications.

The image object allows you to import or reference images, manipulate images by script or interactively with the paint tools, and save them out in different formats with variable compression options. Support extends to alpha channeled PNG images and animated GIF images. Images can be imported and reused within a stack to create custom or interactive interface elements.



Image

# Players

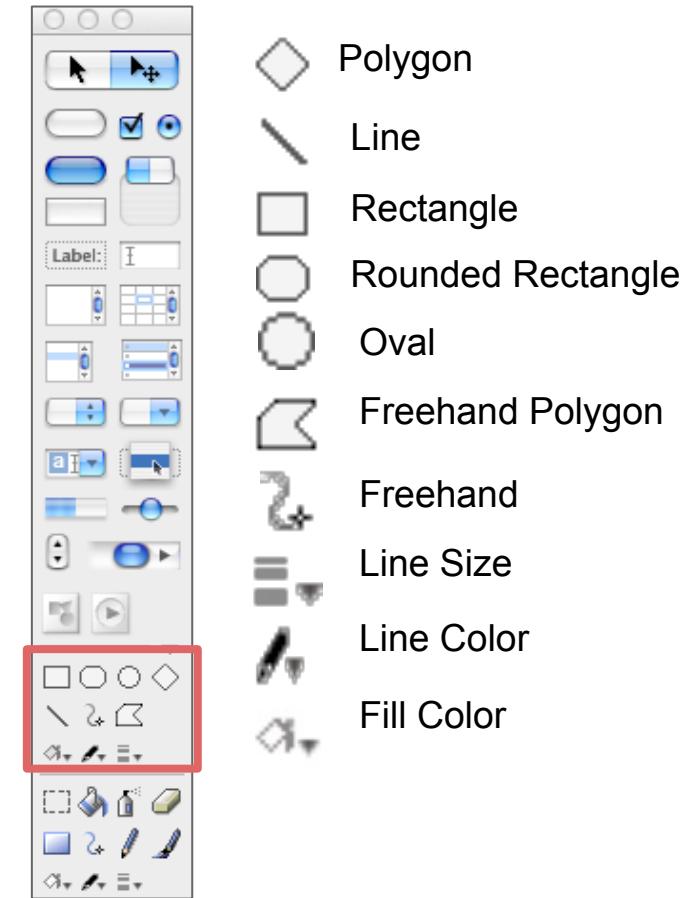
Use the player object to display and interact with any media formats supported by QuickTime. LiveCode allows you to turn on and off tracks within a movie, pan, zoom or change location within a QTVR movie, set callback messages that trigger scripts at specific points in the movie, and stream movies from a server.



Player

# Graphics

LiveCode supports the creation and manipulation of Vector graphics via the graphic tools and by script. LiveCode supports paths with variable fills, gradients, blended and antialiased graphics.





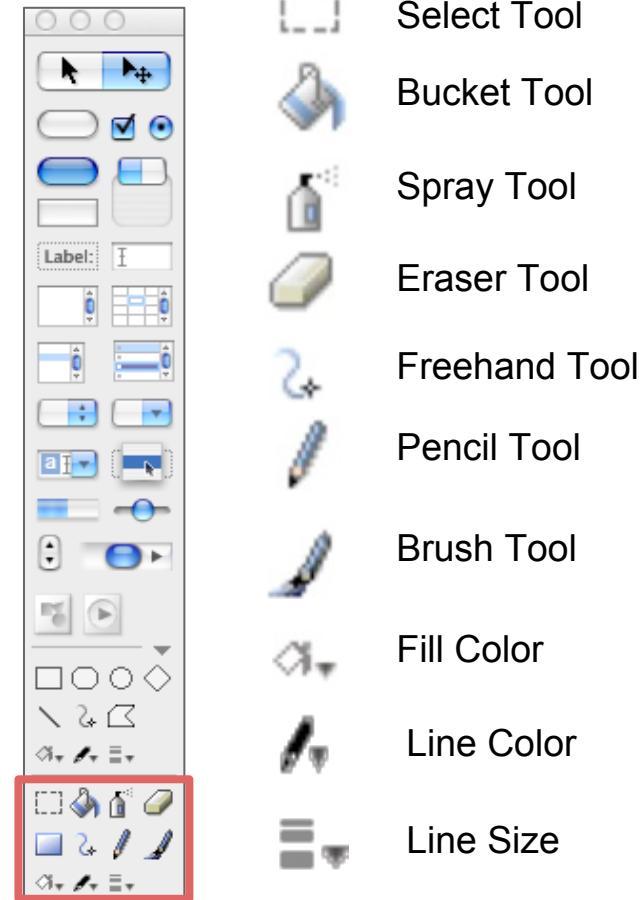
Hello World App

## Paint

LiveCode has a vast array of paint tools that allow some simple image editing within image controls

To use the paint tools you

- 1 Add an image area to the stack.
2. Ensure the paint tools are shown in the Tools Palette by clicking the fold out triangle.
3. Select one of the paint tools.
4. The paint tool can be used within the image area.





Hello World App

# Data Grid

The LiveCode Data Grid enables you to integrate powerful tables and forms into your LiveCode projects. Data grids combine LiveCode groups and behaviors to provide you with a simple, yet flexible means of displaying your data in just about any way you want.

The LiveCode Data Grid is a remarkable and versatile control. We will shortly create a basic data grid and look at a small portion of its functionality.

**LIVE CODE**  
LESSONS

---

## [LiveCode Data Grid](#)

[Introduction](#)

[Data Grid Fundamentals](#)

[Example: Creating a List of People](#)

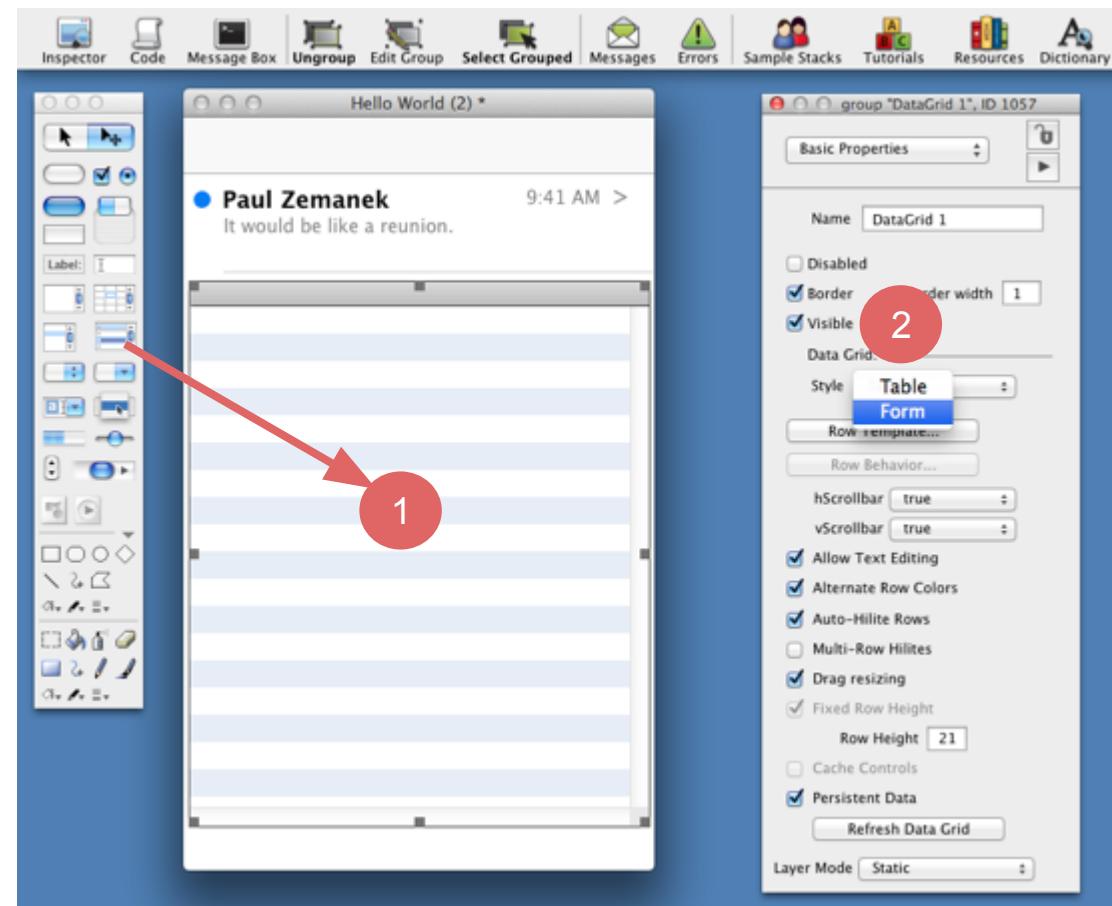
[Working With Data Grid Forms](#)



**Data Grid**

# Let's learn about the Data Grid Control

1. First add a grid control from the Tool Palette.
2. Open the Property Inspector for it (this can be accessed by double-clicking on the control, from the toolbar, from the Object menu and from context sensitive menus).



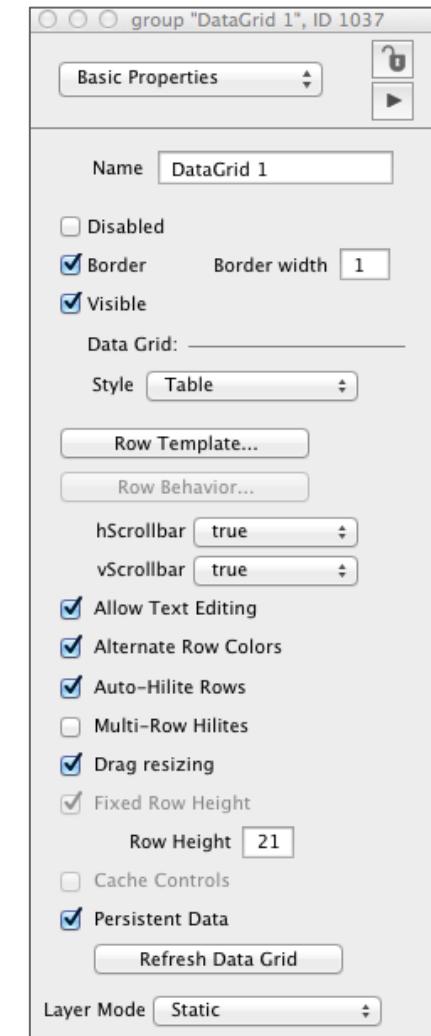


# Say Hello to the...Property Inspector

The last two days, you have been setting values on properties of our controls in your Hello World App. The Property Inspector allows you to view and edit the properties for any selected object. Properties control how an object looks and some aspects of a control's behavior.

To recap, the Inspector can be accessed by double clicking on a selected object, from the toolbar, from the Object menu and from context sensitive menus.

Let's learn more...

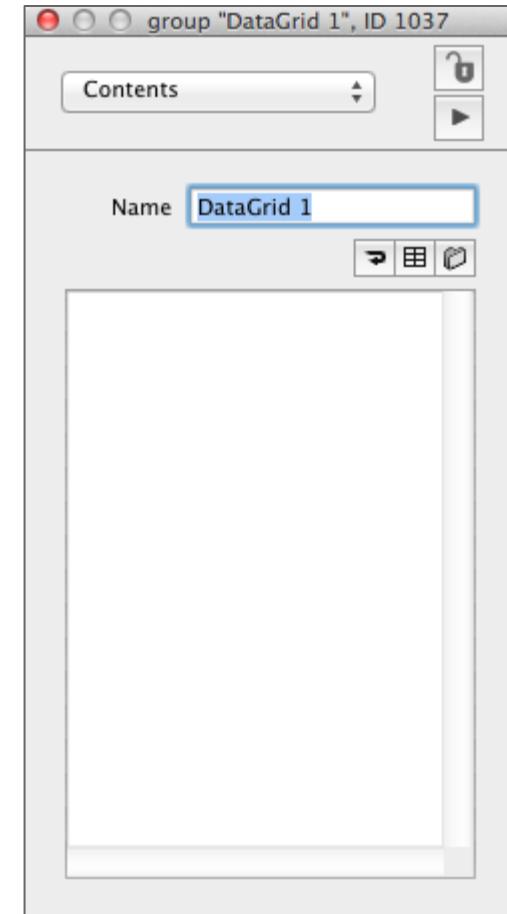




Hello World App

# Contents

The contents section of the property inspector will show the textual contents of the current selected object. If this is a text field then this will be the contents of the field. In the case of a data grid Table you can see the contents of each cell here.

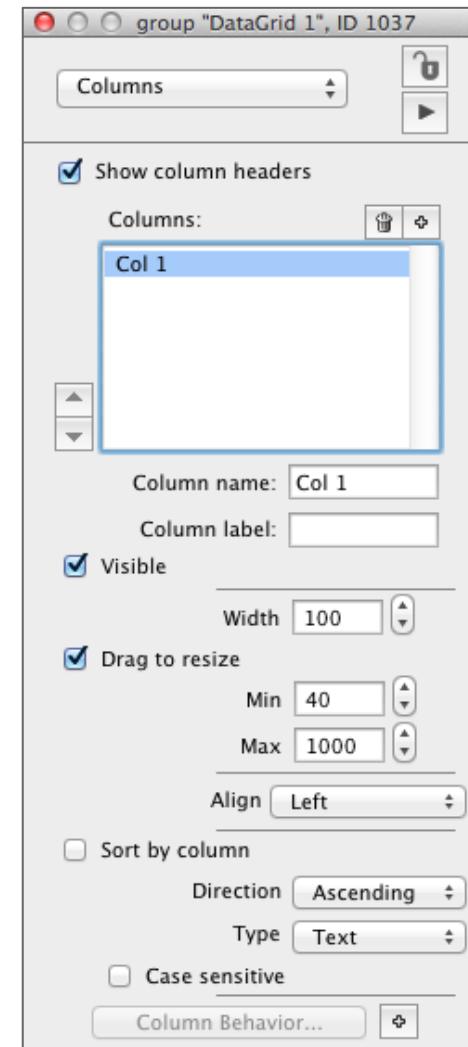




# Columns

A column, in the context of a table style Data Grid, can be seen as an individual group for a specific data set within the grid.

An example could be an address book with a different users details (columns could be Name, Last Name, Address and Phone number).



The screenshot shows the "DataGrid 1" configuration dialog in the LiveCode editor. At the top, there's a dropdown menu labeled "Columns" with up and down arrows, and a lock/unlock icon. Below it is a checked checkbox for "Show column headers". A "Columns:" section contains a single column named "Col 1", which is highlighted with a blue border. To the right of "Col 1" are delete and add buttons. Below the column list are two input fields: "Column name: Col 1" and "Column label: ". Underneath these are several checkboxes: "Visible" (checked), "Drag to resize" (checked), and "Sort by column" (unchecked). For "Drag to resize", there are input fields for "Width" (100), "Min" (40), and "Max" (1000), each with up and down arrows. There are also alignment options ("Align Left") and case sensitivity settings ("Case sensitive" unchecked). At the bottom are "Direction" (set to "Ascending"), "Type" (set to "Text"), and a "Column Behavior..." button.



LIVE CODE

Hello World App

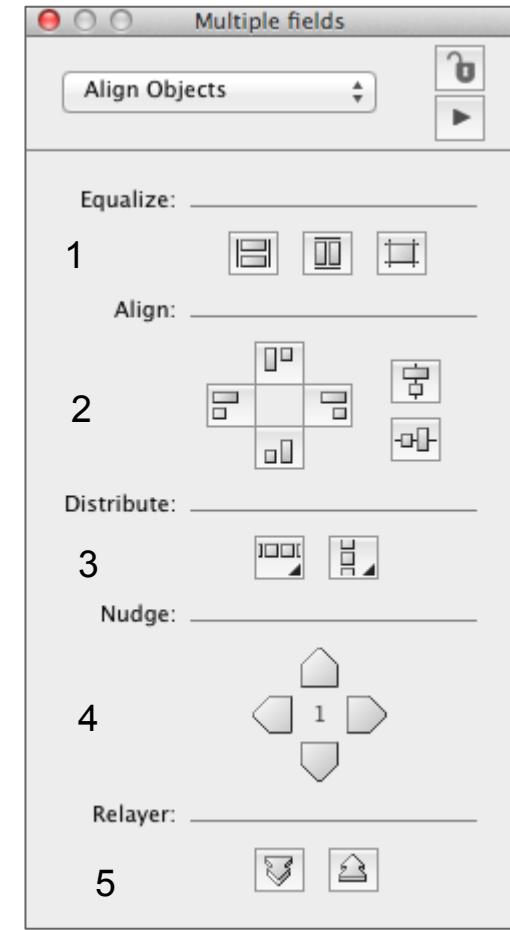
# Align Objects

*\*Note: The Align Objects option ONLY appears if you have multi objects selected on your card.*

To open the Align Objects Inspector, select multiple objects by dragging over them, then open the Inspector and choose Align Objects from the menu at the top of the Inspector. The Align Objects pane will automatically be displayed if you select multiple objects of different types.

There are a number of other options under 'Align Objects'

1. Make the widths and/or height of objects equal.
2. Align objects six different ways.
3. Space objects equally (horizontally or vertically).
4. Nudge objects one pixel at a time.
5. Change the layer of controls.



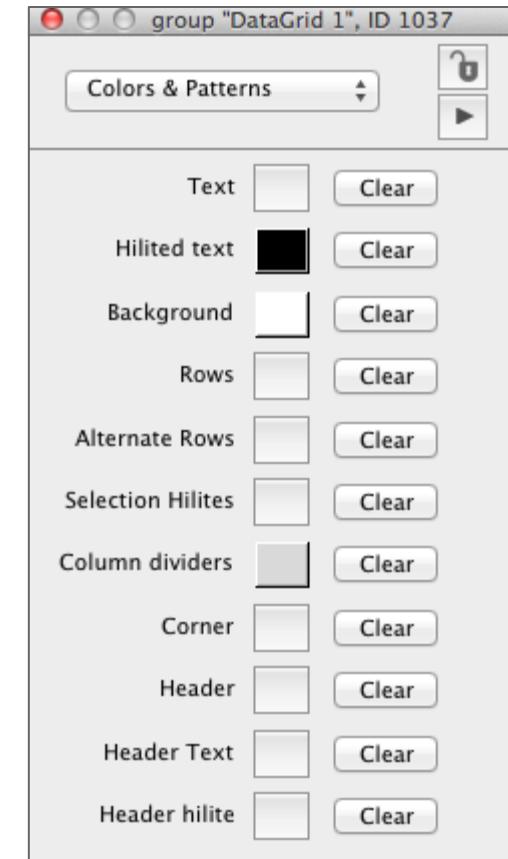


## Hello World App

# Colors & Patterns

This section allows you to change the various colors of the selected LiveCode control. Each item deals with a specific color. Some examples for adatagrid are:

- Text is the default color of the text
- Hilited text is the color of the text when there is a selection
- Row is the color of every odd row
- Alternate row is the color of each even row





Hello World App

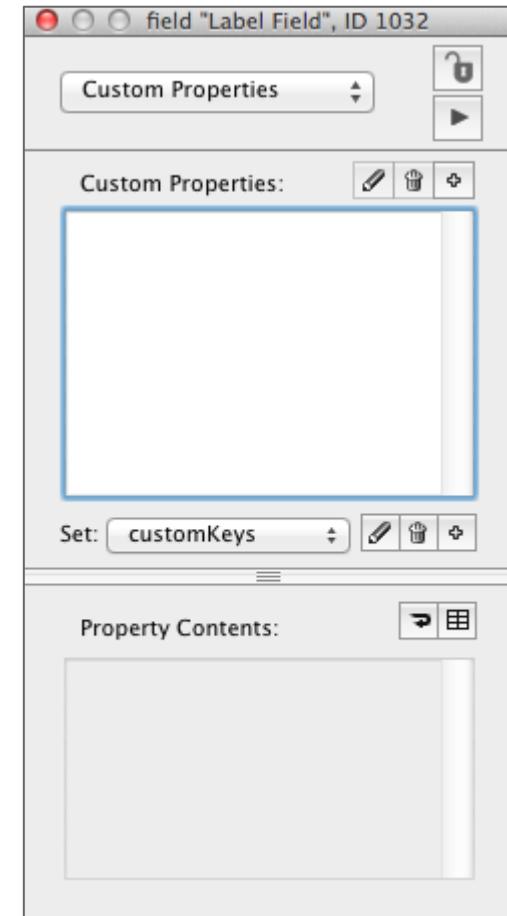
## Custom Properties

A custom property is a property that you create for an object, in addition to its built-in properties. You can define custom properties for any object, and use them to store any kind of data. You can create as many custom properties for an object as you want.

Use a custom property when you want to:

- associate data with a specific object
- save the data with the object in the stack file
- access the data quickly

[More on Custom Properties Here](#)





LIVE CODE

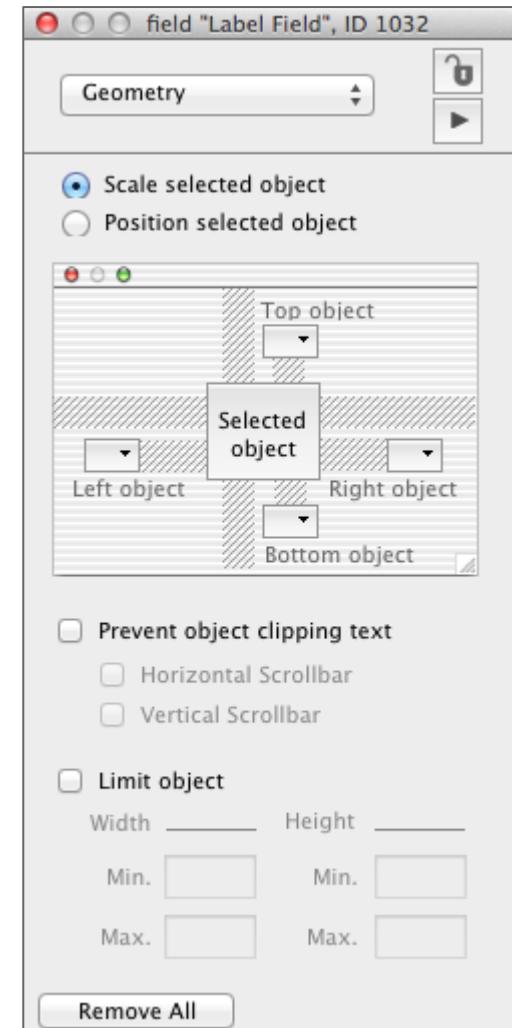
Hello World App

# Geometry

The geometry manager is one way to manage the assets of your stack when the user resizes a stack window.

It is often the case that stacks need to be resizable particularly when they may contain a variable volume of content. We'll come back to this later in the course.

[Geometry Manager Lesson](#)



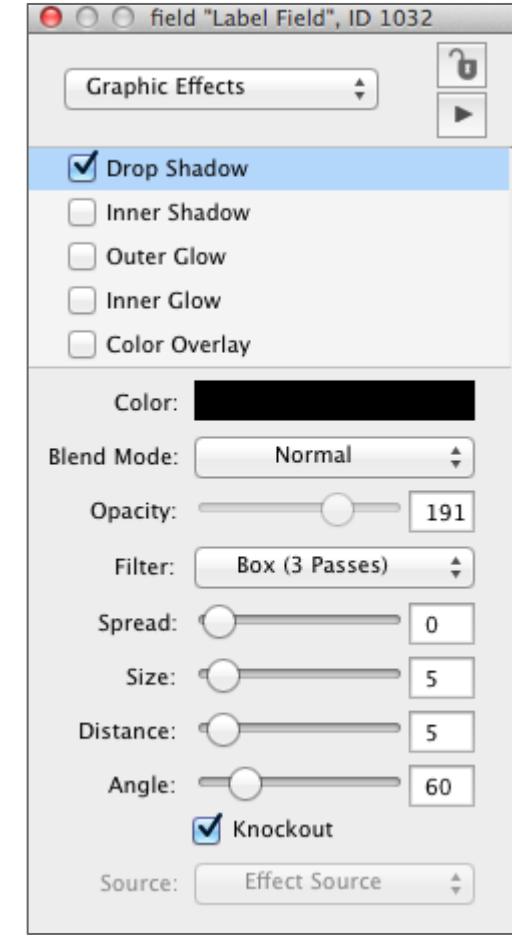


Hello World App

# Graphic Effect

Graphic effects are one way to make your LiveCode controls more aesthetically pleasing. The graphics effects available are some of the most common ones found in many popular photo editing applications.

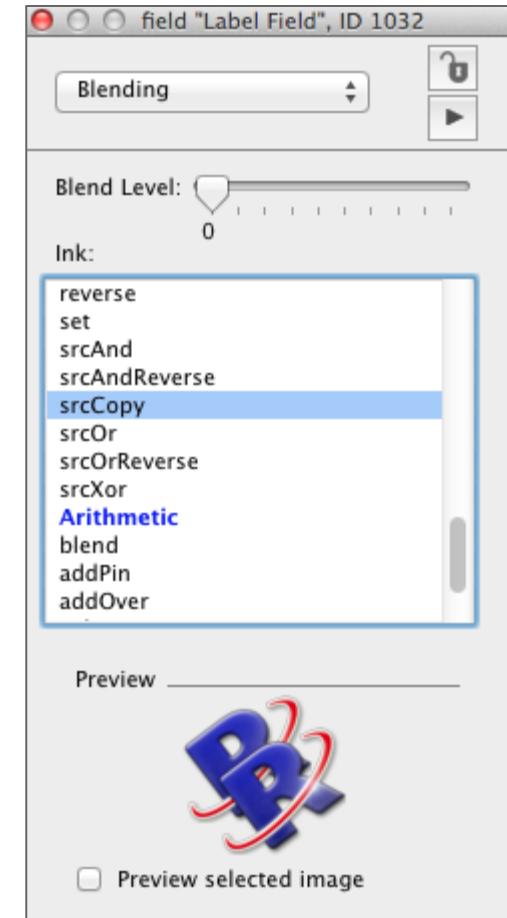
## Using Graphic Effects





# Blending

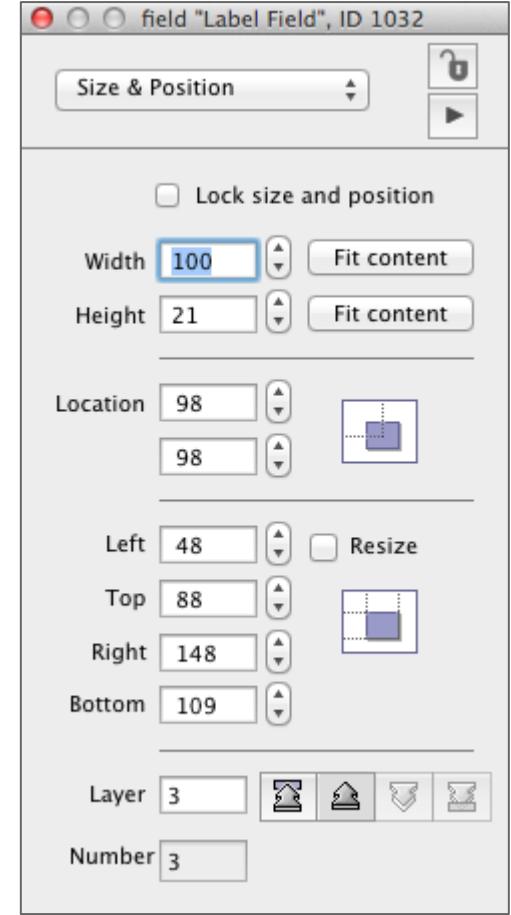
Blend modes determine how an object's colors combine with the colors of the pixels underneath the object to determine how the object's color is displayed. To set the blend mode of an object, use the *Blending* pane in the *Inspector* or set the object's ink property. All objects in LiveCode support blend modes, with the exception of stacks.





## Size & Position

This allows for fine granular control over the size and position of a selected object.



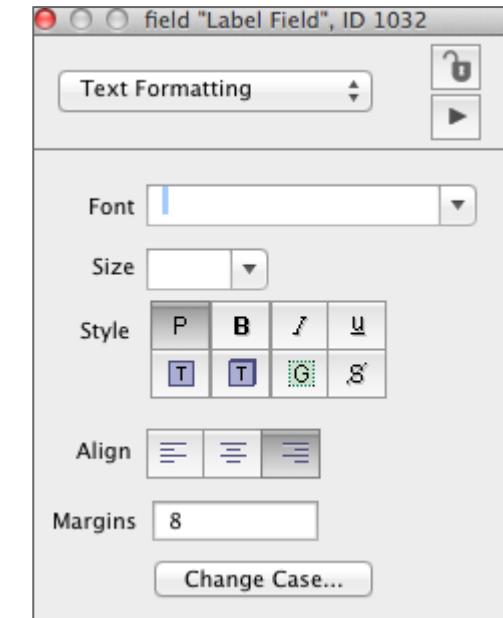
The screenshot shows the 'Size & Position' dialog for an object named 'Label Field', ID 1032. The dialog includes fields for Width (100), Height (21), Location (98, 98), Left (48), Top (88), Right (148), Bottom (109), Layer (3), and Number (3). It also features a 'Lock size and position' checkbox, 'Fit content' buttons, and icons for resize, move, lock, and protect.

Setting	Value	Unit
Width	100	
Height	21	
Location	98	
Left	48	
Top	88	
Right	148	
Bottom	109	
Layer	3	
Number	3	



# Text Formatting

This allows you to set the various Text based properties for a LiveCode object. You are able to select from the various in-built system fonts, set the size of the text, give the text some styling (e.g bold), Align the text and set the margins of the text.





LIVE CODE

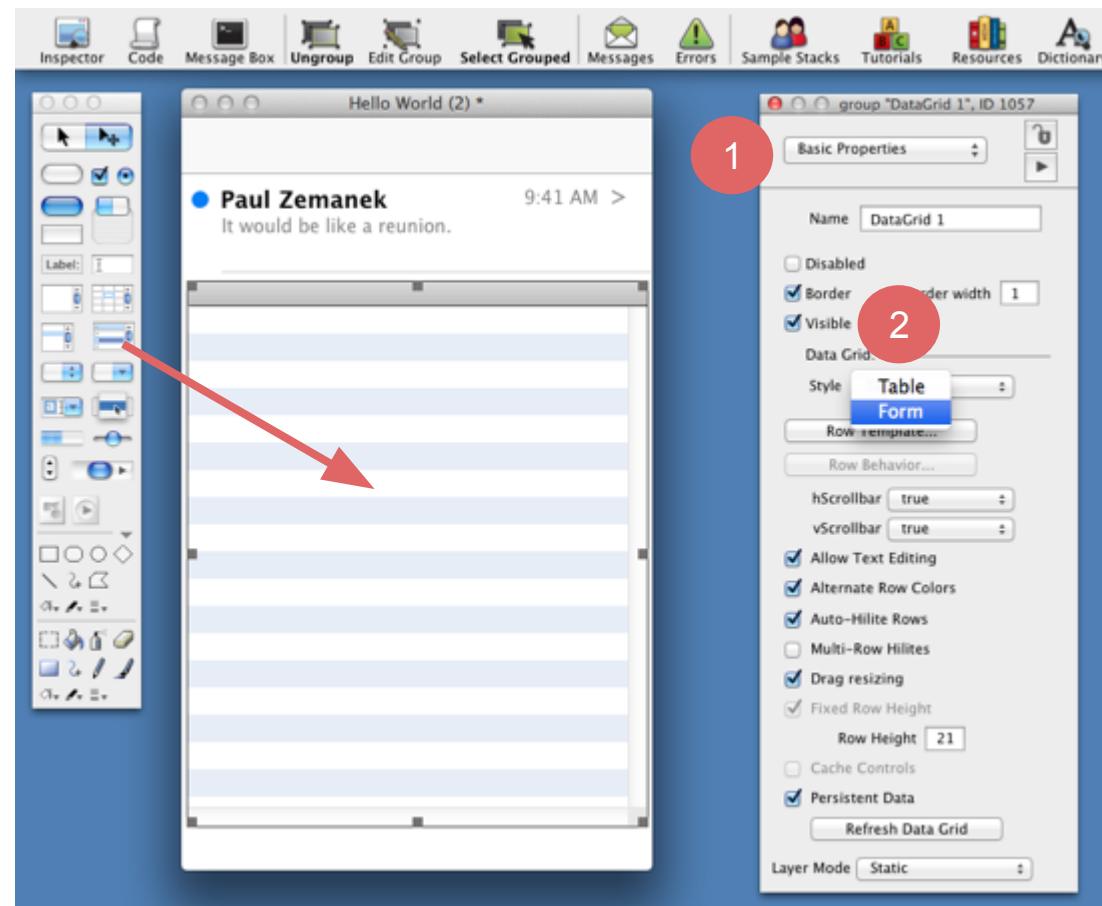
Hello World App

# Let's set a Data Grid property

1. Make sure you have added the grid properly and have selected Basic Properties.
2. The property we will set first is Style. This will tell LiveCode to format the grid in the style of a Table or a Form.

**Table:** The grid will be a series of rows and columns.

**Form:** The grid will be series of a Template which is defined by you.





# LIVE CODE

## Hello World App

# Table vs. Form

A data grid table shows information in a grid, like a spreadsheet. A data grid form shows information in rows, with data item in cols as shown.

Today we want to use the Form style. As you can see, the Form allows us to use a template to style the data. So for each row, LiveCode will use a template you have designed and populate the template with that rows data.

Let's take a two different views of how the template process works..

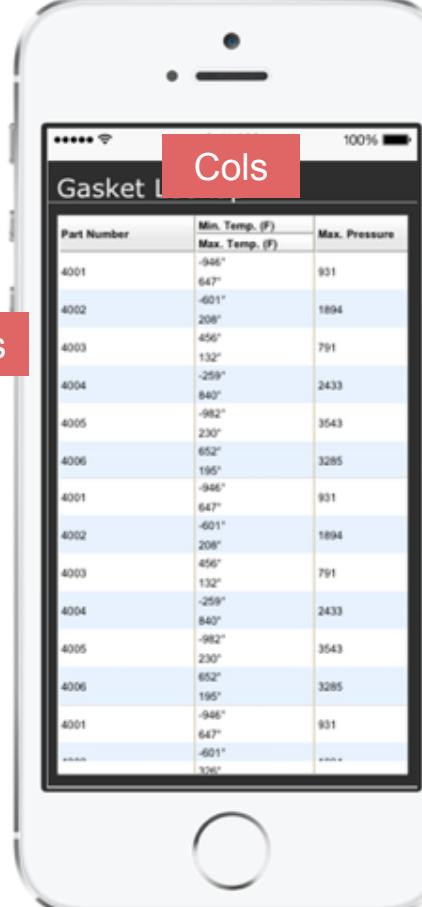
### Table Grid

The screen shows a table titled "Gasket L". It has three columns: "Part Number", "Min. Temp. (F)", and "Max. Pressure". The data is as follows:

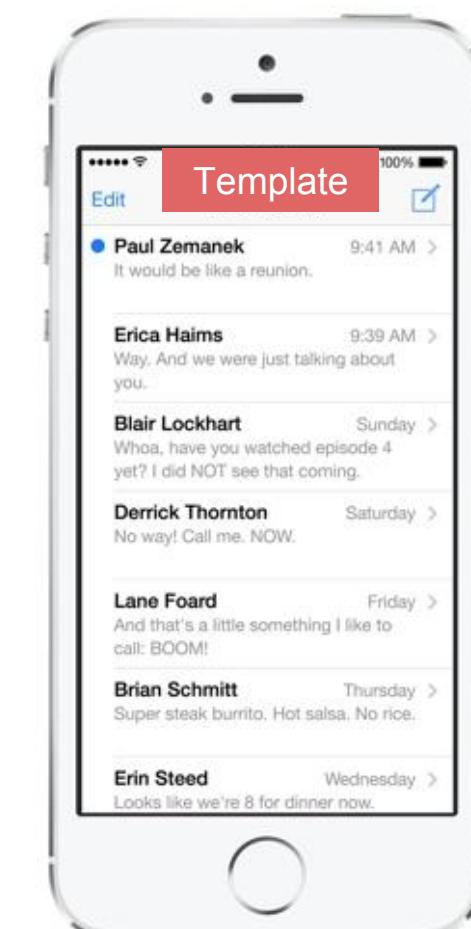
Part Number	Min. Temp. (F)	Max. Pressure
4001	-945°	931
4002	647°	
4003	-601°	1894
4004	208°	
4005	456°	791
4006	132°	
4007	-259°	2433
4008	840°	
4009	-982°	3543
4010	230°	
4011	652°	3285
4012	195°	
4013	-946°	931
4014	647°	
4015	-601°	1894
4016	208°	
4017	456°	791
4018	132°	
4019	-259°	2433
4020	840°	
4021	-982°	3543
4022	230°	
4023	652°	3285
4024	195°	
4025	-946°	931
4026	647°	
4027	-601°	1894
4028	208°	
4029	456°	791
4030	132°	
4031	-259°	2433
4032	840°	
4033	-982°	3543
4034	230°	
4035	652°	3285
4036	195°	
4037	-946°	931
4038	647°	
4039	-601°	1894
4040	208°	
4041	456°	791
4042	132°	
4043	-259°	2433
4044	840°	
4045	-982°	3543
4046	230°	
4047	652°	3285
4048	195°	
4049	-946°	931
4050	647°	
4051	-601°	1894
4052	208°	
4053	456°	791
4054	132°	
4055	-259°	2433
4056	840°	
4057	-982°	3543
4058	230°	
4059	652°	3285
4060	195°	
4061	-946°	931
4062	647°	
4063	-601°	1894
4064	208°	
4065	456°	791
4066	132°	
4067	-259°	2433
4068	840°	
4069	-982°	3543
4070	230°	
4071	652°	3285
4072	195°	
4073	-946°	931
4074	647°	
4075	-601°	1894
4076	208°	
4077	456°	791
4078	132°	
4079	-259°	2433
4080	840°	
4081	-982°	3543
4082	230°	
4083	652°	3285
4084	195°	
4085	-946°	931
4086	647°	
4087	-601°	1894
4088	208°	
4089	456°	791
4090	132°	
4091	-259°	2433
4092	840°	
4093	-982°	3543
4094	230°	
4095	652°	3285
4096	195°	
4097	-946°	931
4098	647°	
4099	-601°	1894
40100	208°	
40101	456°	791
40102	132°	
40103	-259°	2433
40104	840°	
40105	-982°	3543
40106	230°	
40107	652°	3285
40108	195°	
40109	-946°	931
40110	647°	
40111	-601°	1894
40112	208°	
40113	456°	791
40114	132°	
40115	-259°	2433
40116	840°	
40117	-982°	3543
40118	230°	
40119	652°	3285
40120	195°	
40121	-946°	931
40122	647°	
40123	-601°	1894
40124	208°	
40125	456°	791
40126	132°	
40127	-259°	2433
40128	840°	
40129	-982°	3543
40130	230°	
40131	652°	3285
40132	195°	
40133	-946°	931
40134	647°	
40135	-601°	1894
40136	208°	
40137	456°	791
40138	132°	
40139	-259°	2433
40140	840°	
40141	-982°	3543
40142	230°	
40143	652°	3285
40144	195°	
40145	-946°	931
40146	647°	
40147	-601°	1894
40148	208°	
40149	456°	791
40150	132°	
40151	-259°	2433
40152	840°	
40153	-982°	3543
40154	230°	
40155	652°	3285
40156	195°	
40157	-946°	931
40158	647°	
40159	-601°	1894
40160	208°	
40161	456°	791
40162	132°	
40163	-259°	2433
40164	840°	
40165	-982°	3543
40166	230°	
40167	652°	3285
40168	195°	
40169	-946°	931
40170	647°	
40171	-601°	1894
40172	208°	
40173	456°	791
40174	132°	
40175	-259°	2433
40176	840°	
40177	-982°	3543
40178	230°	
40179	652°	3285
40180	195°	
40181	-946°	931
40182	647°	
40183	-601°	1894
40184	208°	
40185	456°	791
40186	132°	
40187	-259°	2433
40188	840°	
40189	-982°	3543
40190	230°	
40191	652°	3285
40192	195°	
40193	-946°	931
40194	647°	
40195	-601°	1894
40196	208°	
40197	456°	791
40198	132°	
40199	-259°	2433
40200	840°	
40201	-982°	3543
40202	230°	
40203	652°	3285
40204	195°	
40205	-946°	931
40206	647°	
40207	-601°	1894
40208	208°	
40209	456°	791
40210	132°	
40211	-259°	2433
40212	840°	
40213	-982°	3543
40214	230°	
40215	652°	3285
40216	195°	
40217	-946°	931
40218	647°	
40219	-601°	1894
40220	208°	
40221	456°	791
40222	132°	
40223	-259°	2433
40224	840°	
40225	-982°	3543
40226	230°	
40227	652°	3285
40228	195°	
40229	-946°	931
40230	647°	
40231	-601°	1894
40232	208°	
40233	456°	791
40234	132°	
40235	-259°	2433
40236	840°	
40237	-982°	3543
40238	230°	
40239	652°	3285
40240	195°	
40241	-946°	931
40242	647°	
40243	-601°	1894
40244	208°	
40245	456°	791
40246	132°	
40247	-259°	2433
40248	840°	
40249	-982°	3543
40250	230°	
40251	652°	3285
40252	195°	
40253	-946°	931
40254	647°	
40255	-601°	1894
40256	208°	
40257	456°	791
40258	132°	
40259	-259°	2433
40260	840°	
40261	-982°	3543
40262	230°	
40263	652°	3285
40264	195°	
40265	-946°	931
40266	647°	
40267	-601°	1894
40268	208°	
40269	456°	791
40270	132°	
40271	-259°	2433
40272	840°	
40273	-982°	3543
40274	230°	
40275	652°	3285
40276	195°	
40277	-946°	931
40278	647°	
40279	-601°	1894
40280	208°	
40281	456°	791
40282	132°	
40283	-259°	2433
40284	840°	
40285	-982°	3543
40286	230°	
40287	652°	3285
40288	195°	
40289	-946°	931
40290	647°	
40291	-601°	1894
40292	208°	
40293	456°	791
40294	132°	
40295	-259°	2433
40296	840°	
40297	-982°	3543
40298	230°	
40299	652°	3285
40300	195°	

Rows

### Table Grid



### Form Grid



Erica Haims 9:39 AM Way. And we were just talking about you.

Blair Lockhart Sunday Whoa, have you watched episode 4 yet? I did NOT see that coming.

Derrick Thornton Saturday No way! Call me. NOW.

Lane Foard Friday And that's a little something I like to call: BOOM!

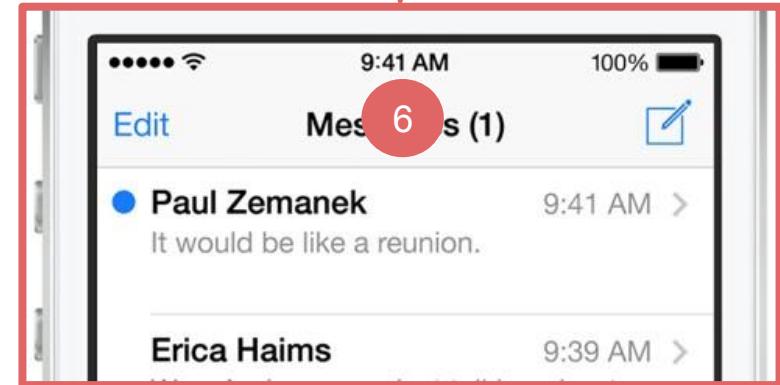
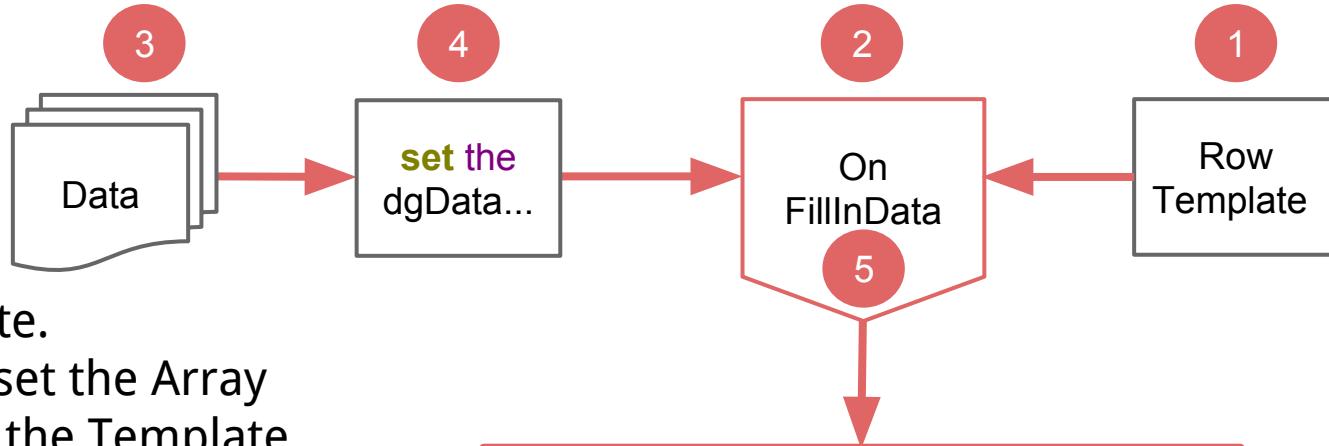
Brian Schmitt Thursday Super steak burrito. Hot salsa. No rice.

Erin Steed Wednesday Looks like we're 8 for dinner now.

## Process Flow View

1. Add a Row Template.
2. Code FillInData to set the Array Data to controls in the Template.
3. Load the data in an array.
4. Set the array to the grid  

```
set the dgData of group "MsgList" to MsgData
```
5. FillInData displays each data item in the array in the list.
6. The new row appears in the Grid.



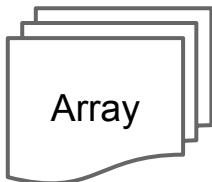


LIVE CODE

Hello World App

# Physical Flow View

3 Get Data



4 Set Data Array to the Grid

set the dgData of group "MsgList" to MsgData

1 Create Row Template

• NameofSender

MessageText

MsgTime >

FillInData

5

2 Code FillInData

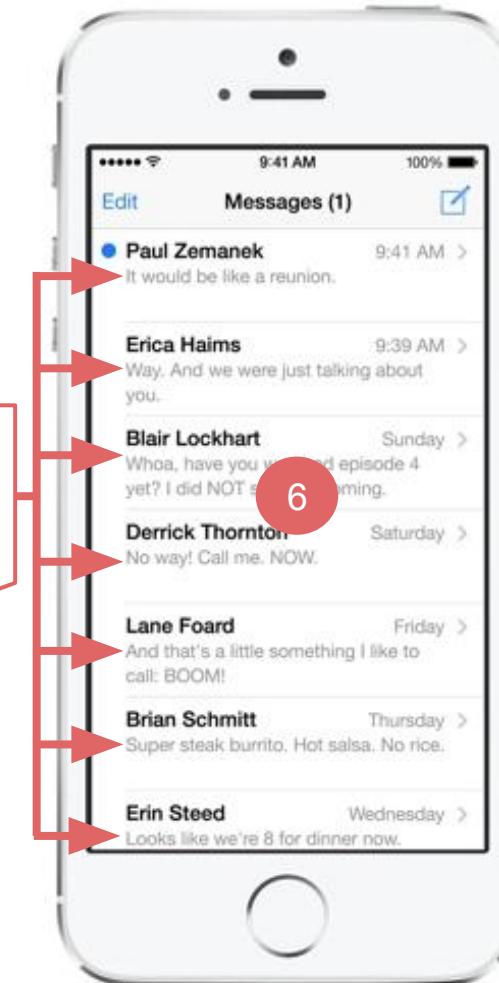
on FillInData pDataArray

  set the text of field "NameofSender" of me to pDataArray["NameofSender"]

...

  set the text of field "MsgTime" of me to pDataArray["MsgTime"]

end FillInData





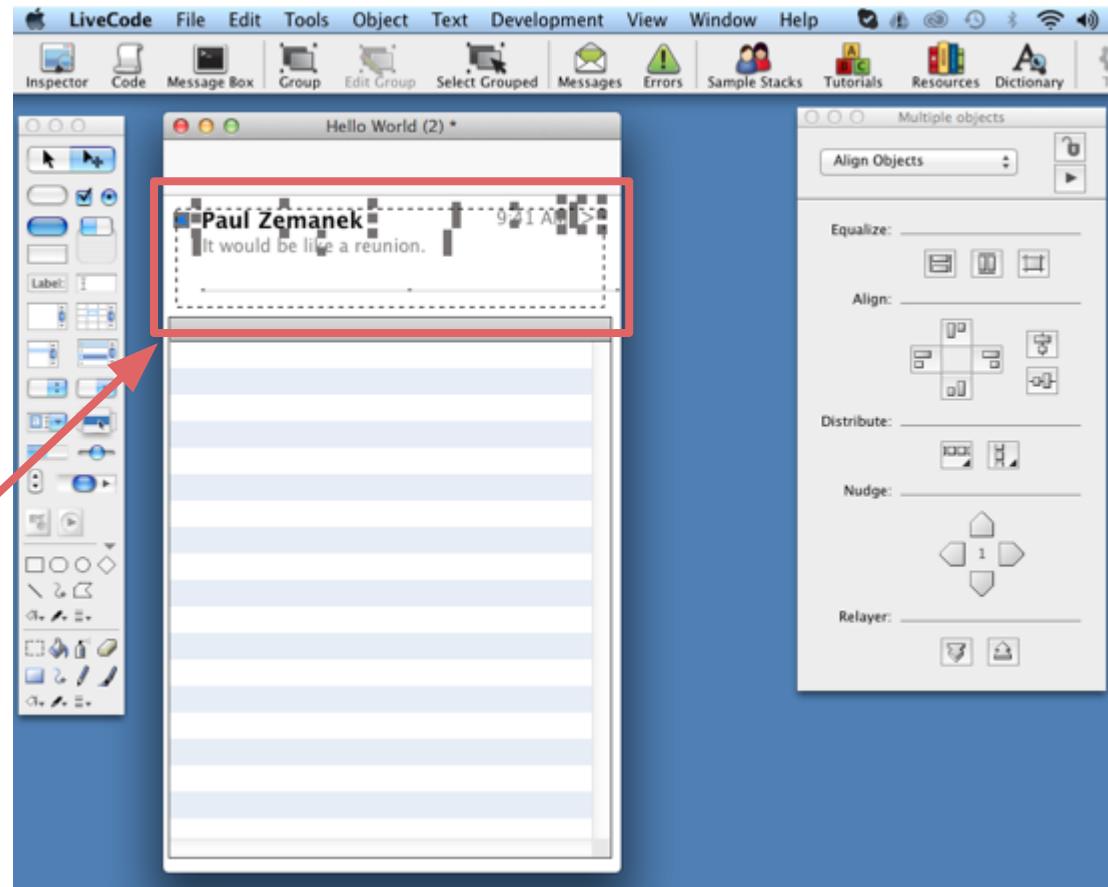
## Hello World App

### Step 1

## Create a Row Template

If you think about it, we have already created our template in the last lesson. Let's save time and effort [always a good idea] and copy our controls.

1. Select all of the controls as you see. Make sure not to select the header or the grid.
2. You should see the selector dark gray boxes on all of the controls.
3. Now copy the controls to the clipboard. Just copy, do not paste yet.

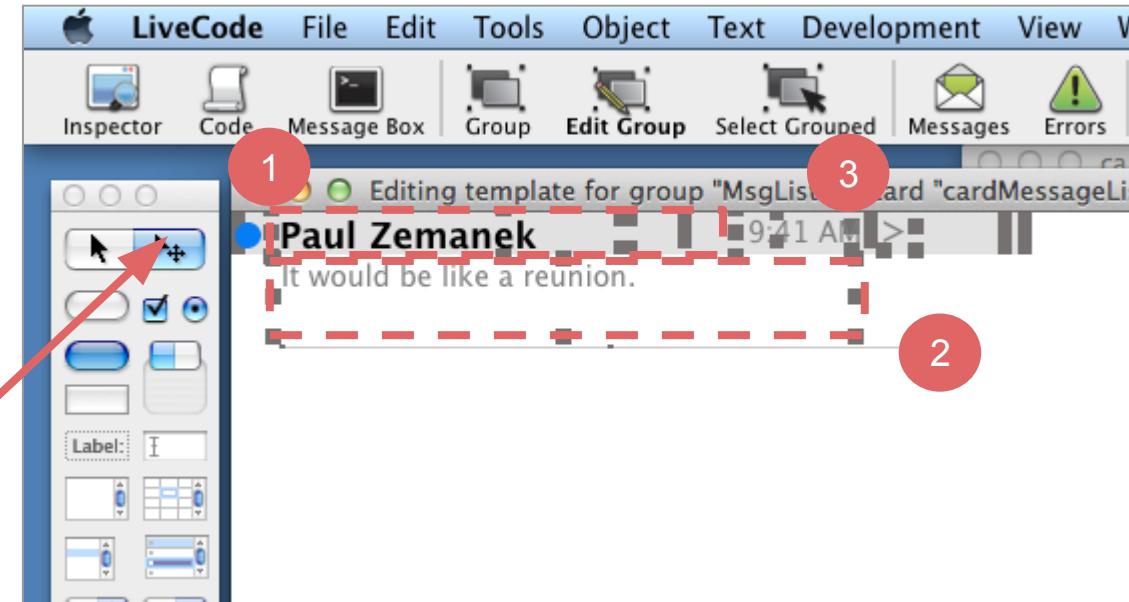


## Step 1

### Create a Row Template

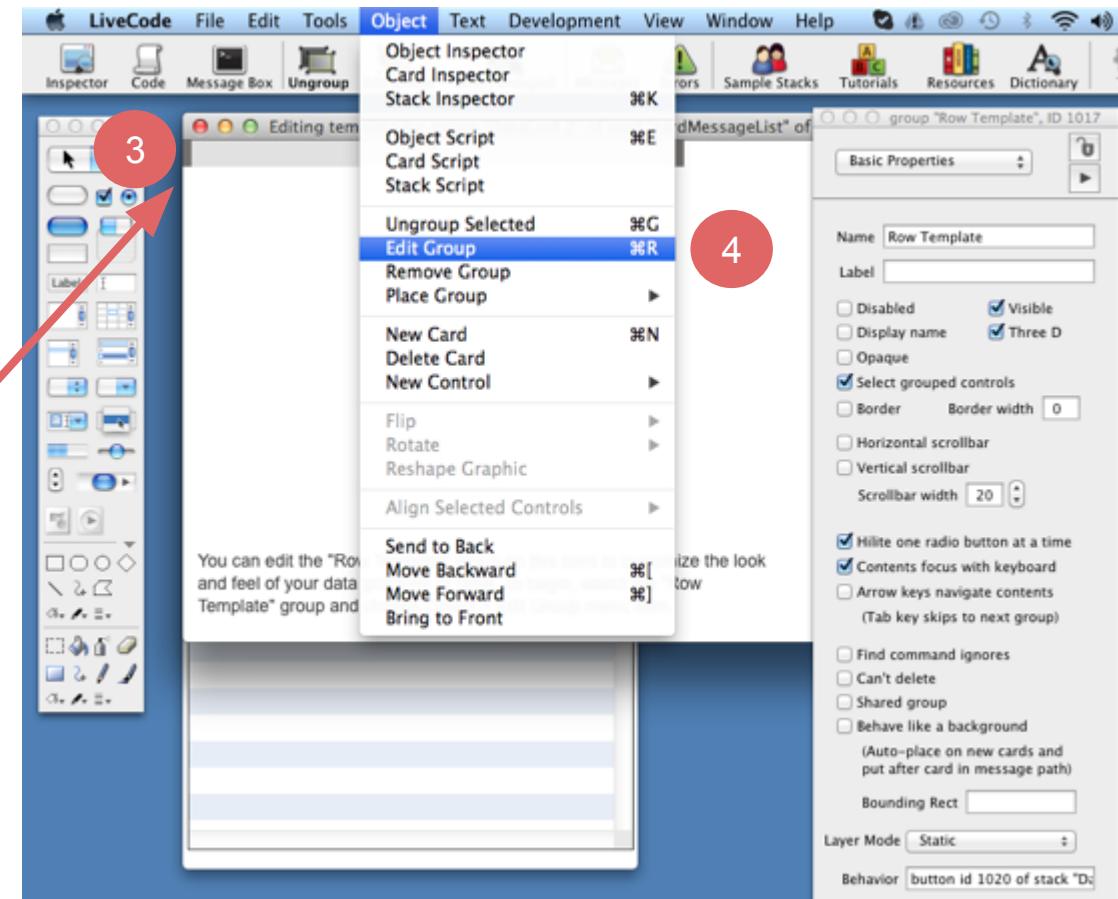
Just a design issue to be careful of. Many forget to size the controls to fill the area desired. You will see that we have sized the Sender Name and the message text to the correct size.

1. Size the Sender Name just before the Message time. Not too close or it will overlap if the day is long
2. The Message text is below the time, but extended to the right alignment of the time. **\*Turn off the Don't Wrap**
3. The time should be aligned right, and the name and message aligned left.



## Step 1 Create a Row Template

1. Now click and select only the data grid open the Property Inspector.
2. Click Row Template button in the Property Inspector.
3. You will see a gray box on the top of the pop-up. Select it. Dark gray boxes will appear to indicate that it is selected.
4. Now select in "Object" in the menu, then "Edit Group".



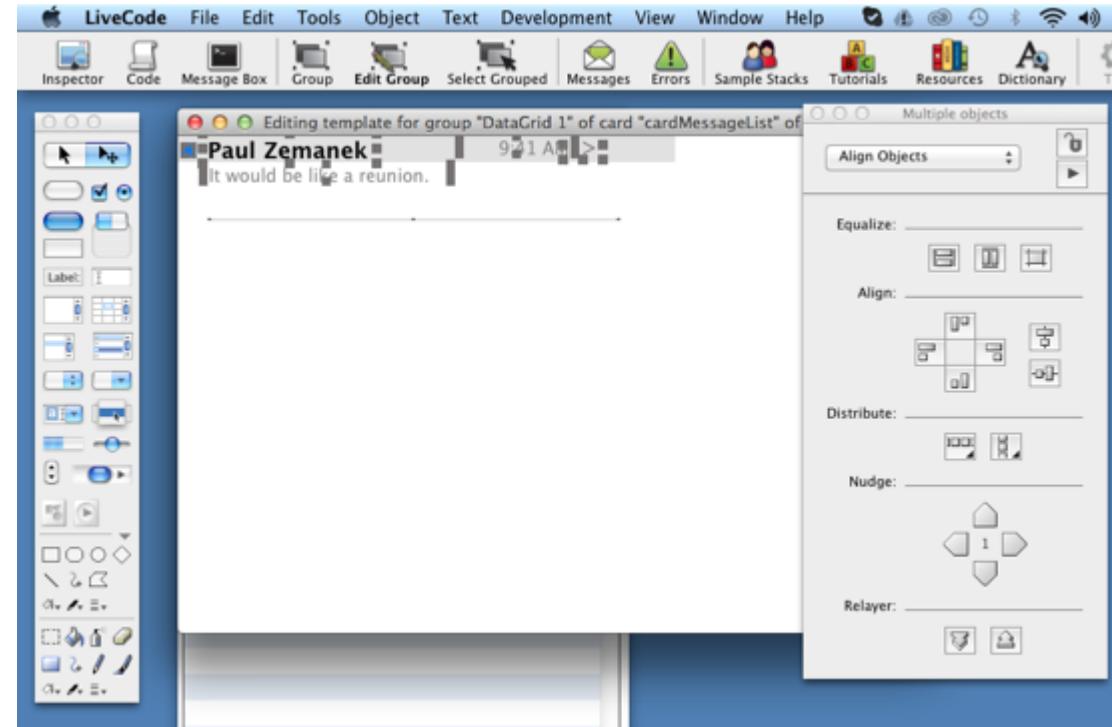


## Step 1

### Create a Row Template

1. Paste your template controls.
2. While they are still all selected, move them to the top left corner as you see on the right.

Congratulations, you are on your way to creating your first row template. Since we are going to set our data to these controls, LiveCode needs to have the name set so it can identify each control.





# LIVE CODE

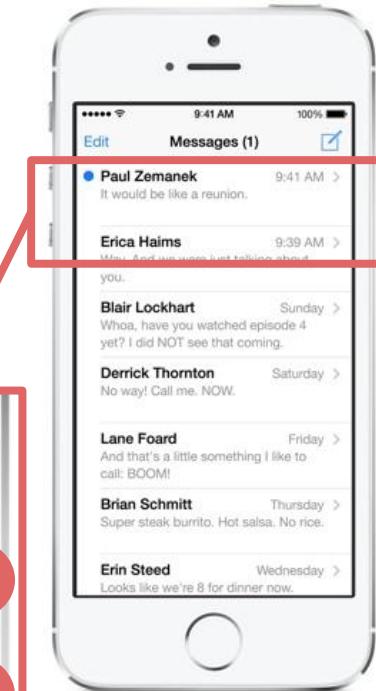
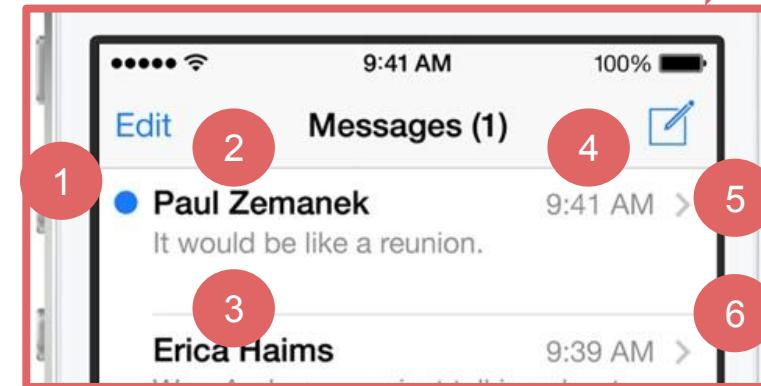
## Hello World App

### Step 1

## Create a Row Template

Let's look at our row template as see what data we will need:

1. **New Message** - Will indicate if there is a new message.
2. **Sender** - The person who sent the message.
3. **Message Text** - The actual message text.
4. **Message Time** - The time of the message.
5. **Navigation Arrow** - No Data
6. **Separator Line** - No Data



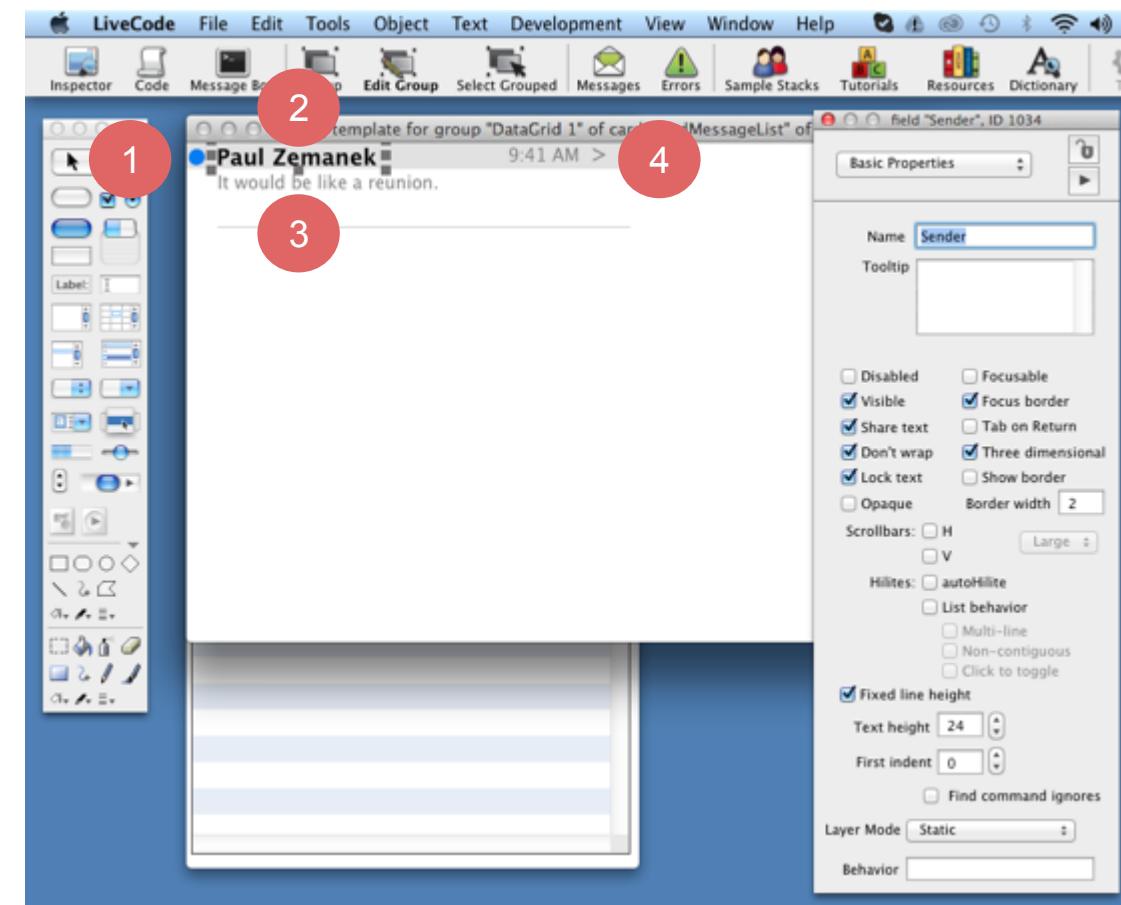


## Step 1

### Create a Row Template

Please select each control and set the name property accordingly. In LiveCode you can have spaces in the Name of controls, but it is good programming practice not to have spaces and just set your name as one word.

1. **NewMessage**
2. **Sender**
3. **MessageText**
4. **MessageTime**



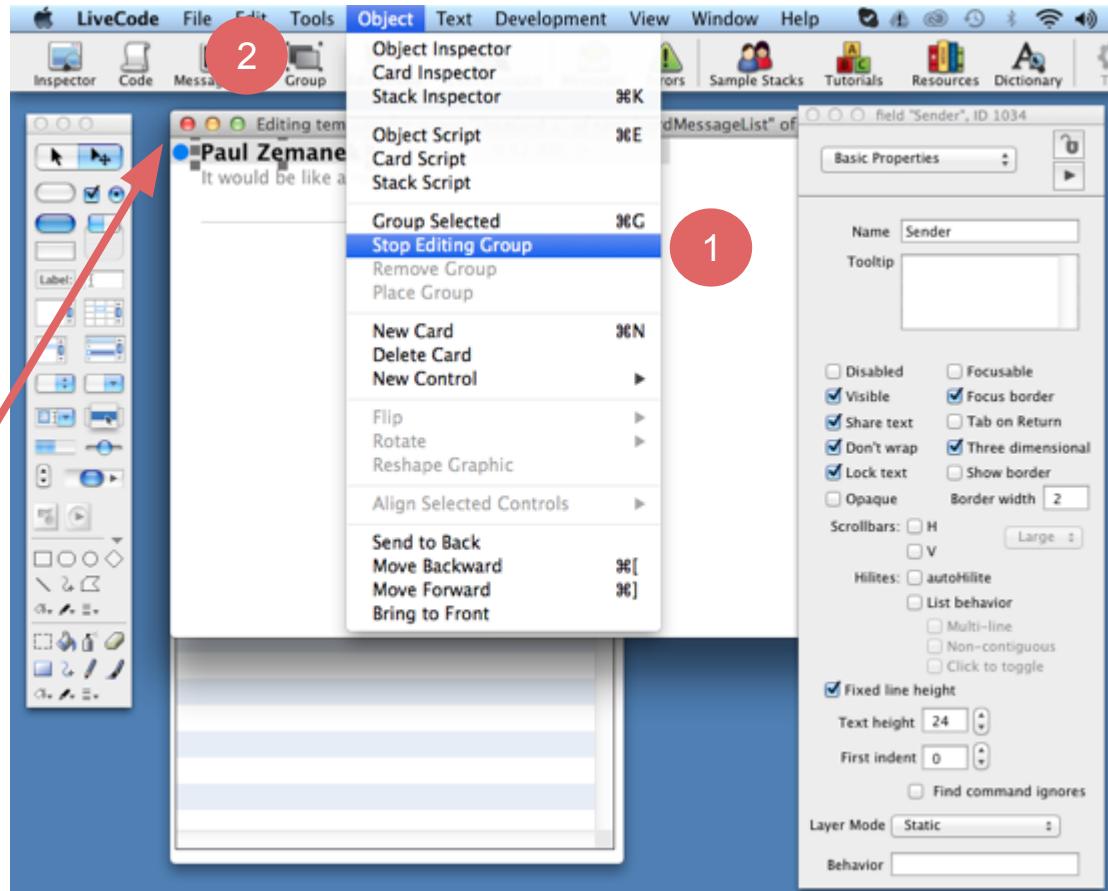
## Step 1 Create a Row Template

The next step is to save your row template.

1. Select the “Stop Editing Group” from the Object menu.

2. Then to complete and save the template, you must close the window. On the Mac, it is the Red circle and on Windows is will be the “x” option.

3. You will be prompted to save - **YOU MUST click the Save option**, or all your changes will be lost.





# LIVE CODE

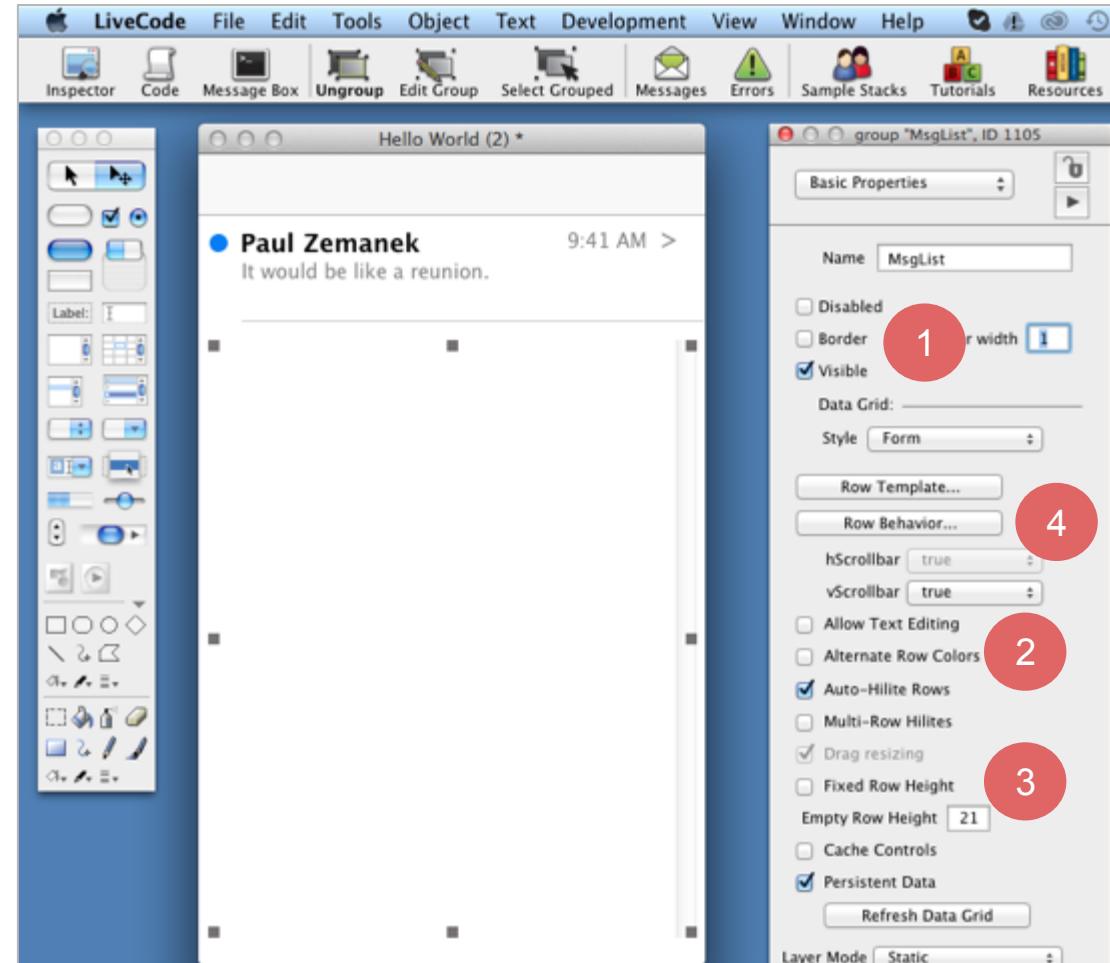
## Hello World App

### Step 1

## Create a Row Template

We now need to set a few properties so LiveCode can style the grid so as to emulate the Message List. The three properties need to be turned off [unchecked the box]. You grid should now look like the image on the right.

1. **Border**
2. **Alternate Row Colors**
3. **Fixed Row Height**
4. Click Row Behavior to start Step 2 in our process.

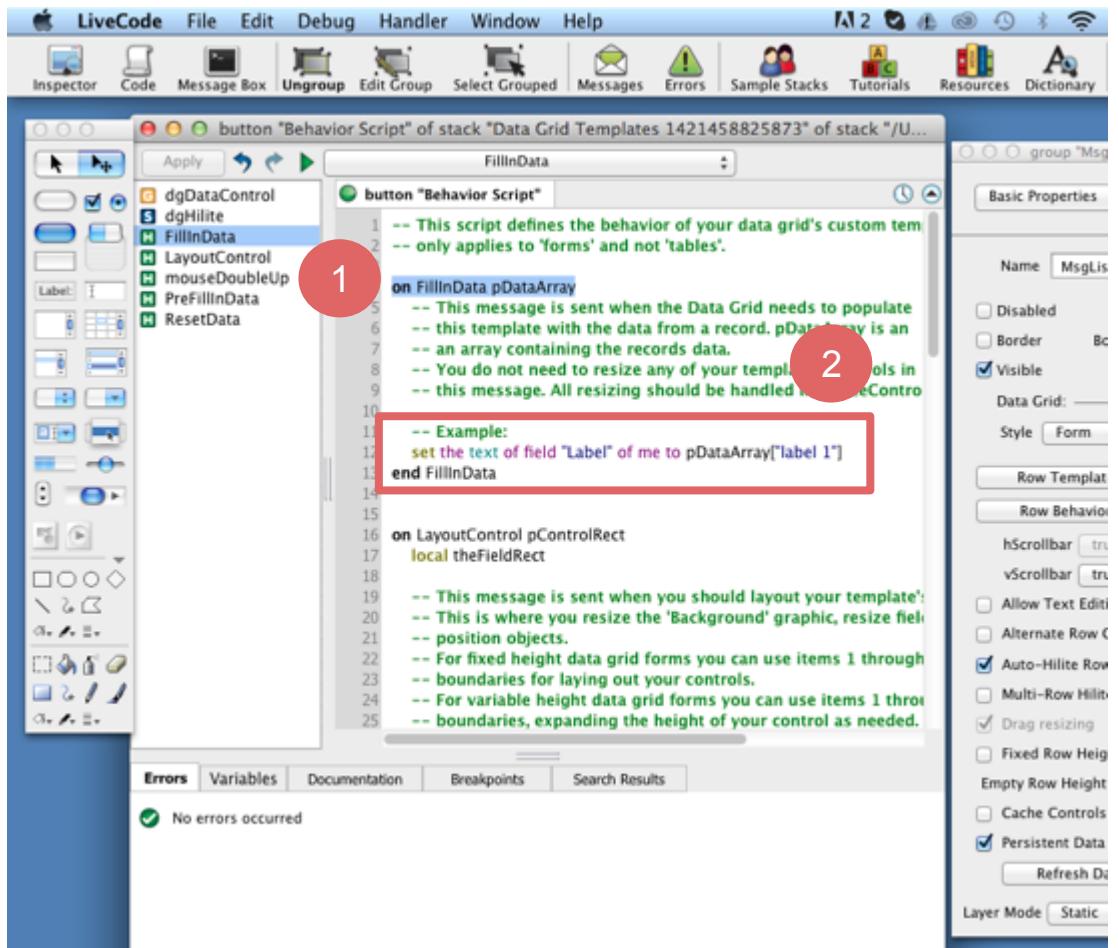


## Step 2

### Code FillInData

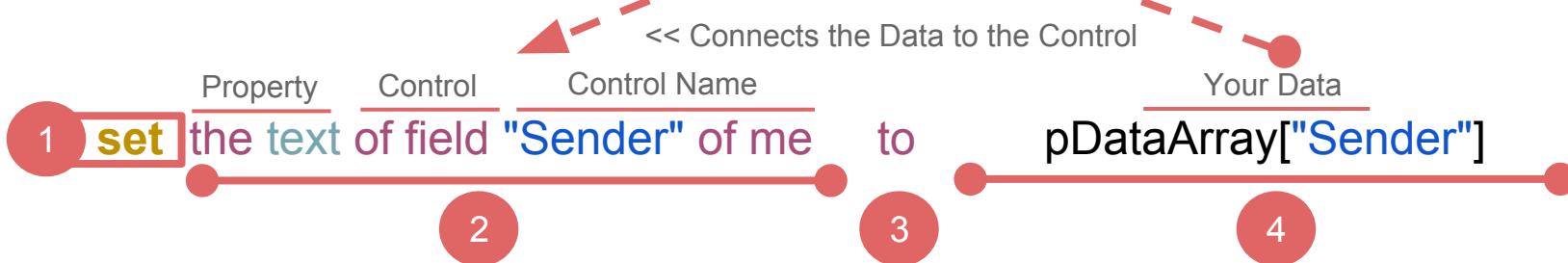
We now need to set a few properties so LiveCode can style the grid so as to emulate the Message List. The three properties need to be turned off [unchecked the box]. You grid should now look like the image on the right.

1. Select FillInData on the left list. This will display and highlight the where we need to code next.
2. Next we need to understand and change the default Example code.





## Step 2 Code FillInData



### Connecting the Row Template Controls with the Data

FillInData is called for each row in your grid. Its job is to connect your data to the control. For example, the first sender is "Paul Zemanek" - FillInData will take that data and set it to the text of field "Sender".

1. First is the command. "Set" tells LiveCode that it is going to set a property
2. What Property of what Control do you want to set or update [\* in FillInData you **MUST** put **of me**]
3. The "to" is the connector. It is a more simple way to say "="
4. Where in your dataArray to get the value from.



## Step 2 Code FillInData

Do you see the connection?

|-----|

|-----|  
Current code in FillInData...

```
| set the text of field "Label" of me to pDataArray["label 1"]
```

| change the code to...

--The values of NewMessage will show the circle if it is new  
-- or will hide the circle if it is not

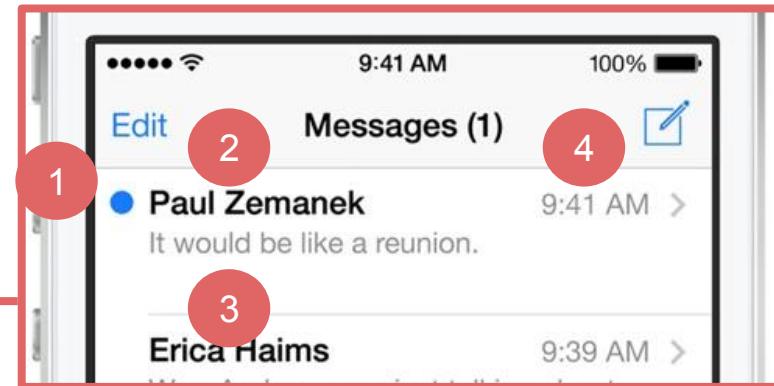
1 set the visible of graphic "NewMessage" of me to pDataArray["newmessage"]

--The rest of the text of the fields are set

2 set the text of field "Sender" of me to pDataArray["Sender"]

3 set the text of field "MessageText" of me to pDataArray["MessageText"]

4 set the text of field "MessageTime" of me to pDataArray["MessageTime"]



Click the Apply Button and you are good to go!



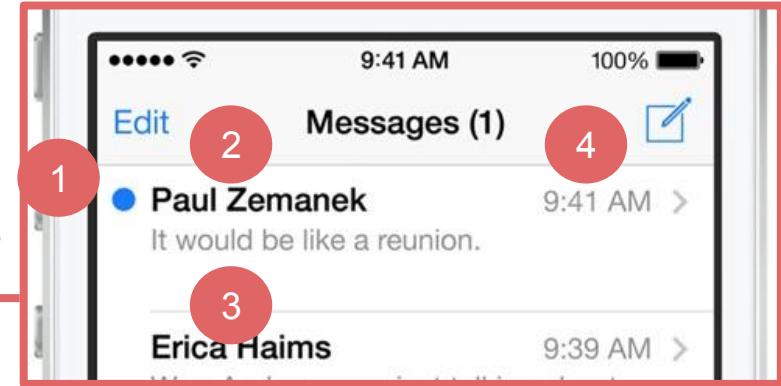
## Step 3

### Get Data

Today is quite a complex lesson, so we wish to make it a little easier for you. We will be covering data in great detail in a future lesson, so for now we provided you a file to copy from. [It will also save you time from typing everything.]

1. Find and open the LoadMessages.txt in today's lesson file.
2. Highlight all of the text and copy it to the clipboard.

Do you see the connection?



- 1 put "true" into MsgData[1]["NewMessage"]
- 2 put "Paul Zemanek" into MsgData[1]["Sender"]
- 3 put "It would be like a reunion." into MsgData[1]["MessageText"]
- 4 put "9:41 AM" into MsgData[1]["MessageTime"]

Take a moment to save your work



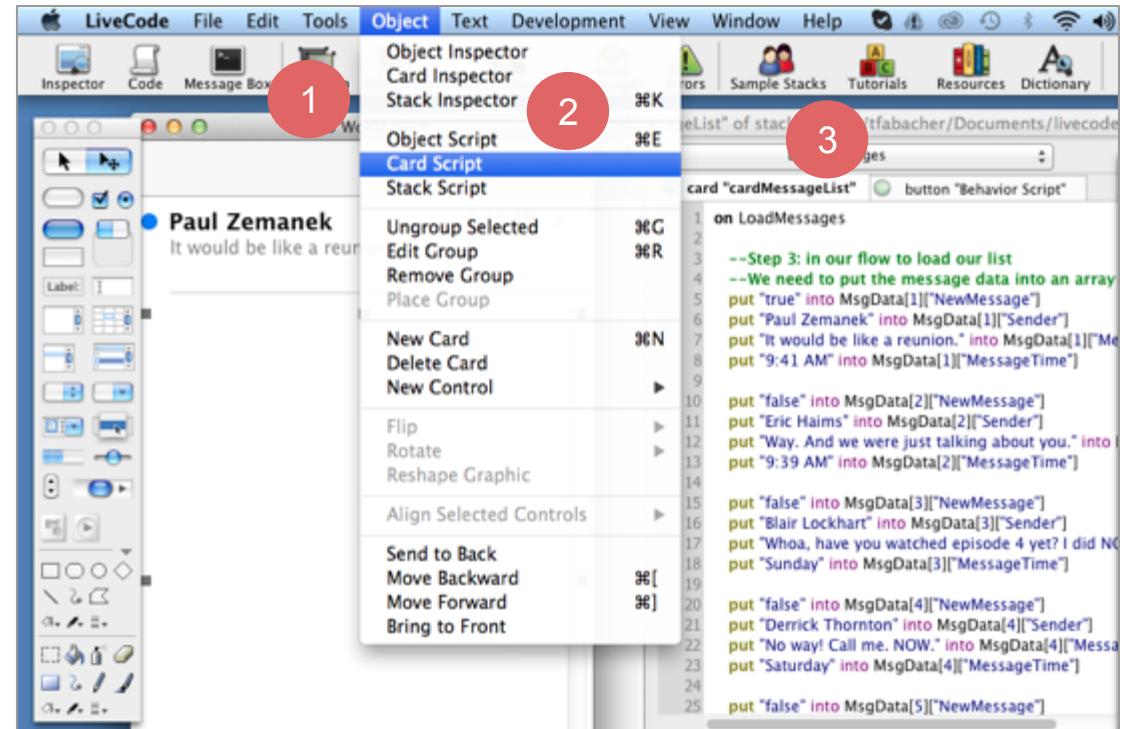
## Step 3 Get Data

New we need to open the code editor and paste in our data.

1. Click on on the stack, so the Object Menu appears.
2. Select Card Script.

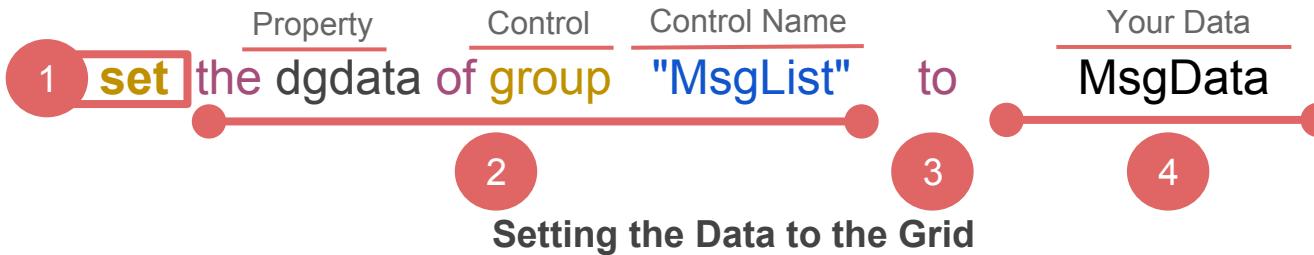
3. Just Paste in the LoadMessages text and LiveCode will automatically do everything needed to import it.

Next we simply have to connect the data to the grid. Luckily this can be done with just 1 line of code in LiveCode.



## Step 4

### Set Data Array to the Grid



We have put in a placeholder where you should add the code to set your data to the grid:

--Step 4: in our flow to load our list  
--Below is the code that will send the data to the grid.

1. First is the command. "Set" tells LiveCode that it is going to set a property
2. We are setting the dgdata property of the grid MsgList.
3. The "to" is the connector. It is a more simple way to say "="
4. We have already built the data above into MsgData. We are just connecting them here.



# LIVE CODE

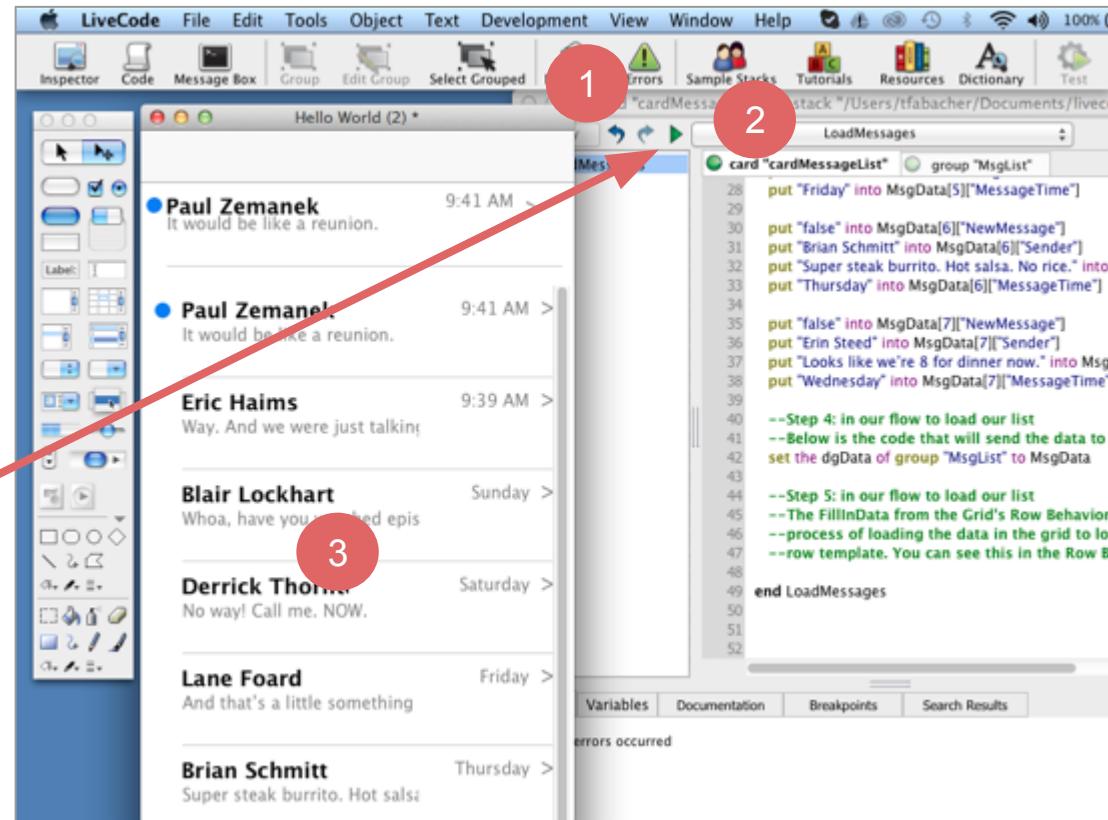
## Hello World App

### Step 5

### Grid calls FillInData

It is finally time for the FillIndata to use the template and data to emulate the iPhone Message List.

1. Click Apply on the Code Editor.
2. Then click the Green “Play” or Continue button. If you get a pop-up just say OK.
3. BINGO! You will witness the LiveCode magic - the list will appear in live.



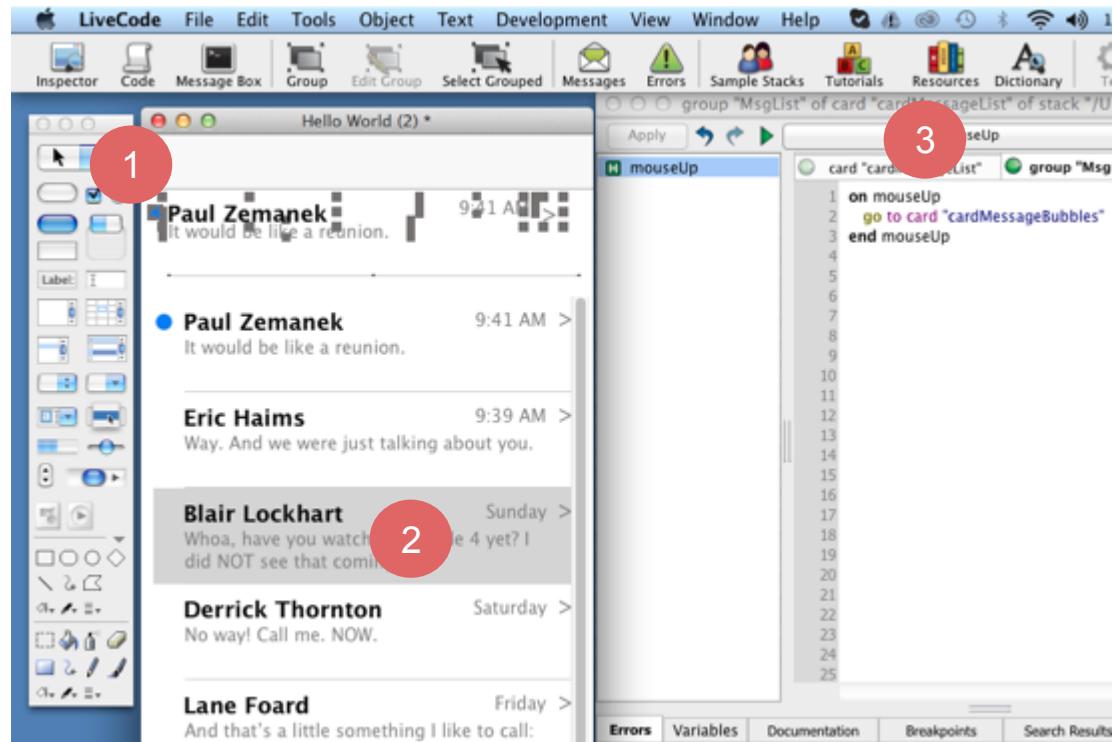


## Step 6 Congratulations!

There is just a few things remaining.

1. Select the old controls [make sure you have not selected the header or the grid] and hit the delete key.
2. Now select the grid and size it to fill in the body area of the App., like your iPhone.
3. Last we need to add the code to go the Message Bubble card. Do this from the Edit Script option on the Grid...click Apply and we are done!

```
on mouseUp
    go to card "cardMessageBubbles"
end mouseUp
```





LIVE CODE

Hello World App

# Congrats on learning: **Controls**

**Don't forget to save your LiveCode Project!**



If you wish to learn more... [Visit LiveCode](#)