LIVE**CODE** (https://livecode.com)   **Platform (https://livecode.com/products/livecode-platform/)**

Resources (https://livecode.com/resources/)   **Pricing (https://livecode.com/products/livecode-platform/pricing/)**

Services (https://livecode.com/services/)   **Blog (https://livecode.com/blog/)**   **login (https://livecode.com/login/)**

Dictionary   Guides   **Lessons**   Courses
(https://livecode.com/resources/api/) (https://livecode.com/resources/guides/) (https://livecode.com/products/learn/)

Objects and Components

This lesson will give you an overview of the objects and components you will use when building LiveCode applications. We will cover the basics of adding objects to your application, using LiveCode to add script to your controls and give you an overview of the inner workings of LiveCode so you can make a start on your own applications.

Starting up LiveCode
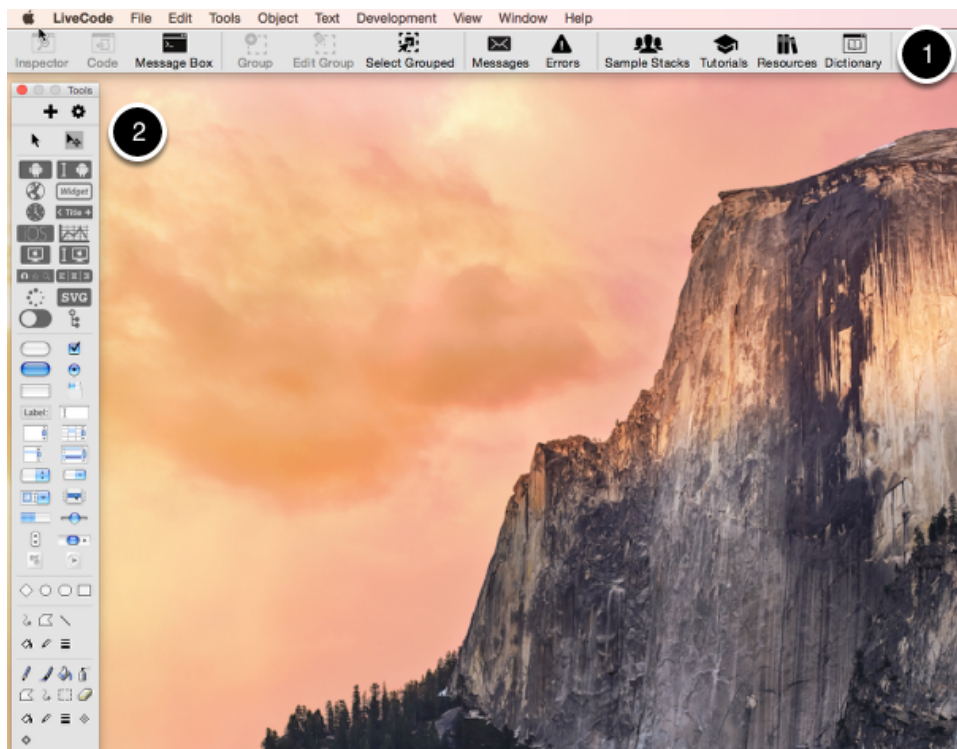
It seems obvious but the first step is to start up LiveCode. When you start LiveCode you will see the menu bar and toolbar at the top of the screen(1). The menu bar and toolbar give you access to all the things you would expect, new files, saving, script editors etc as well as useful resources including Rev Online and the dictionary. We'll go into all the options in more detail as we need them.

The Tools Palette (2) allows you to add controls to your application and switch between "edit" and "run" modes. Again we'll go into these in more detail later.

Stacks and cards
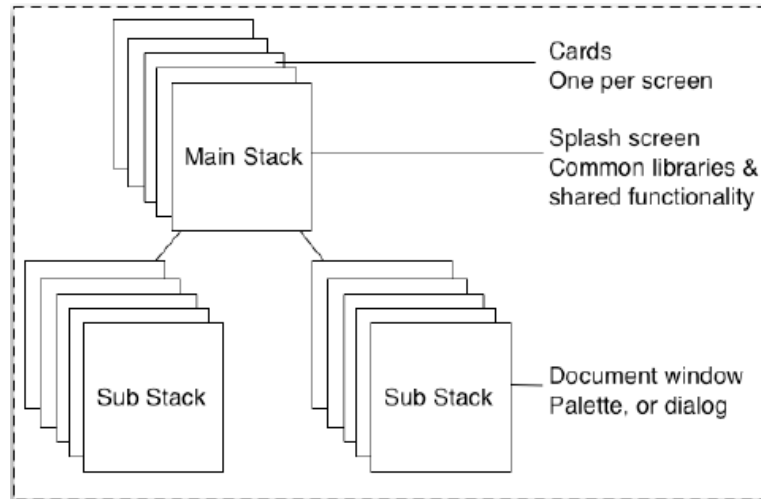
Applications in LiveCode are known as stacks and the separate screens which make these up are called cards. A stack can contain any number of cards and may also contain substacks. The substacks will generally provide specialist actions or implement specific parts of your application.

Your first stack
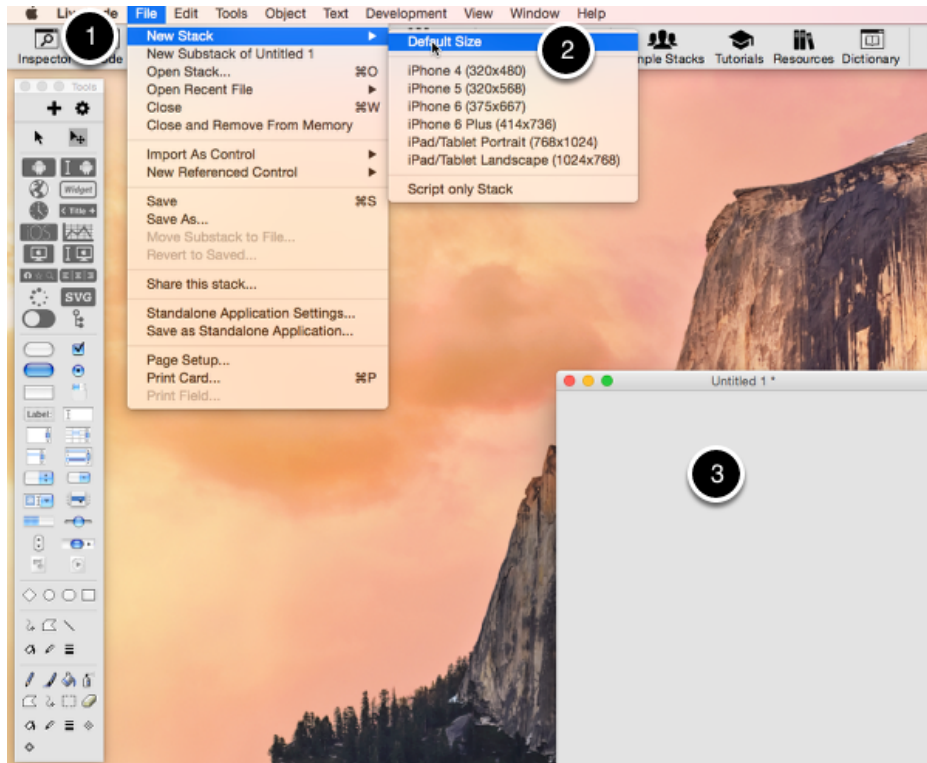
First things first, we need a stack so we can start building our application. Just go to the File menu (1) and select **New Stack/Default Size (2).** We now have a stack (3) containing a single card.

Run and Edit mode
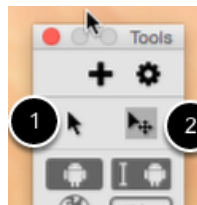
You can use the Tools Palette to switch between **Run**(1) and **Edit**(2) modes while you are working. When in Edit mode objects can be placed on the cards, resized and edited. When in Run mode no editing can take place but buttons etc can be clicked and results observed.

Adding controls to the stack
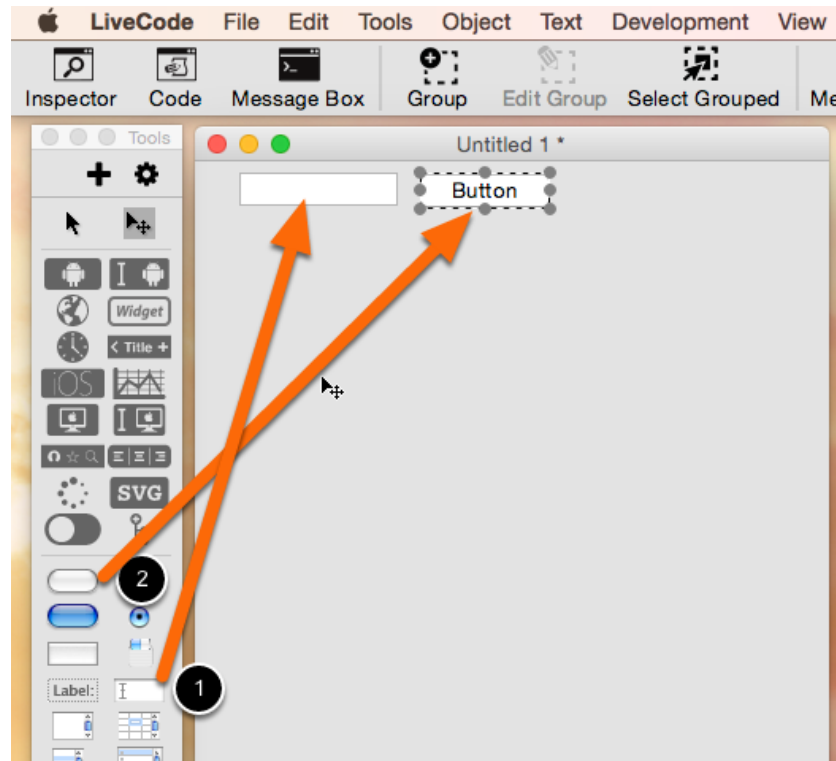
To add controls to the stack you simply drag the type of control you want over from the Tools Palette and place it on your card wherever you want it. Lets add a basic field and button to our card. Just select the field(1) on the Tools Palette and drag it to your card, then do the same with the button(2).
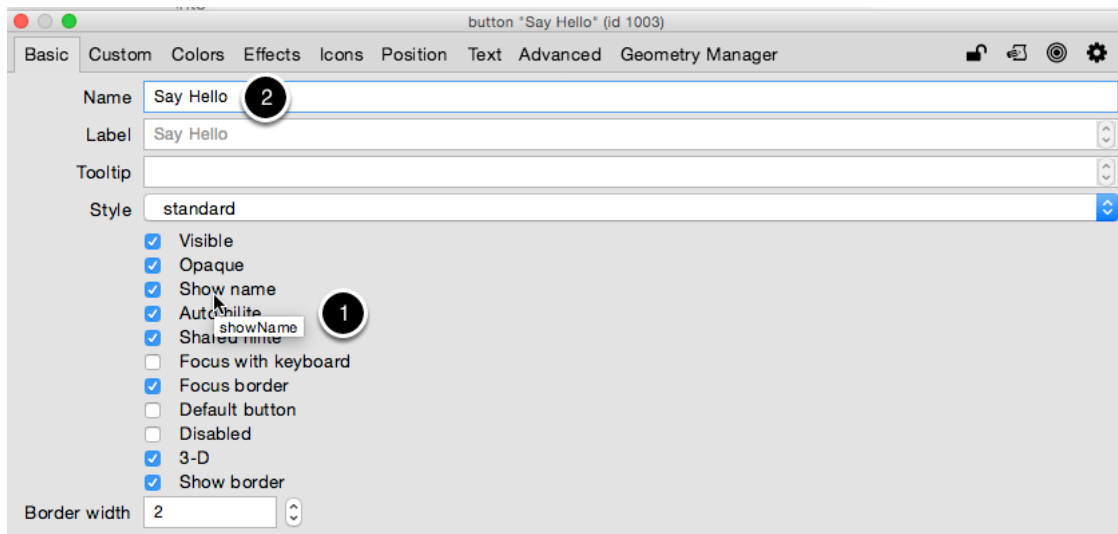
Object properties
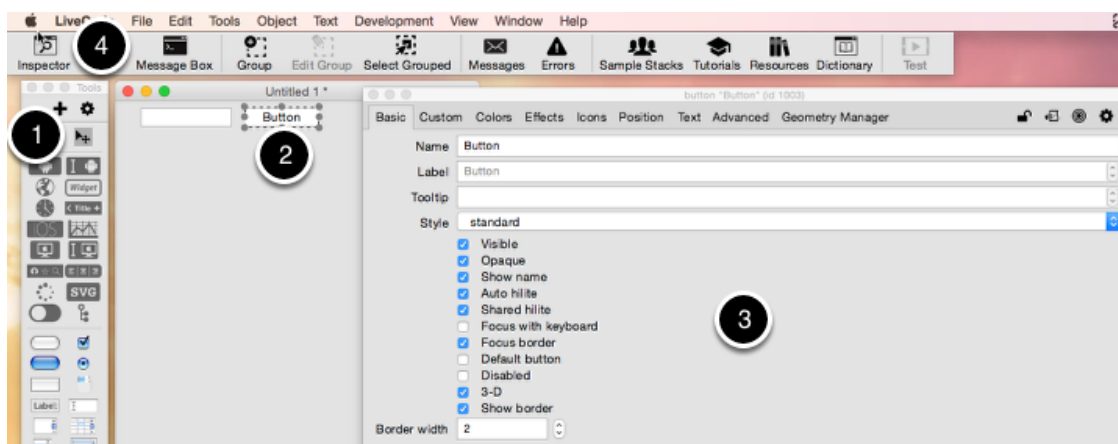
Now that we have a (very simple) interface set up what can we do with it? Every object in LiveCode has associated properties which specify how it looks and behaves. These can be altered when LiveCode is in Edit mode(1) and the easiest way to do this is using the Property Inspector (2). This can be opened by selecting the object(3) and clicking the inspector button on the menu

bar(4). Alternatively you can open the Property Inspector by right clicking the object and selecting **Property Inspector** from the resulting menu or, alternatively, by double clicking the object itself.

The Property Inspector

The Property Inspector displays all of the object's properties, from its name and title to its display settings. You can use the property inspector to change the properties of an object. The best way to find out what a property does is to try changing it and see what happens. Hovering over an option in the Property Inspector will tell you the property name(1) which you can look up in the Dictionary if you want to find out more.

Here I have renamed the button to be called "Say Hello" (2).

Adding scripts to controls
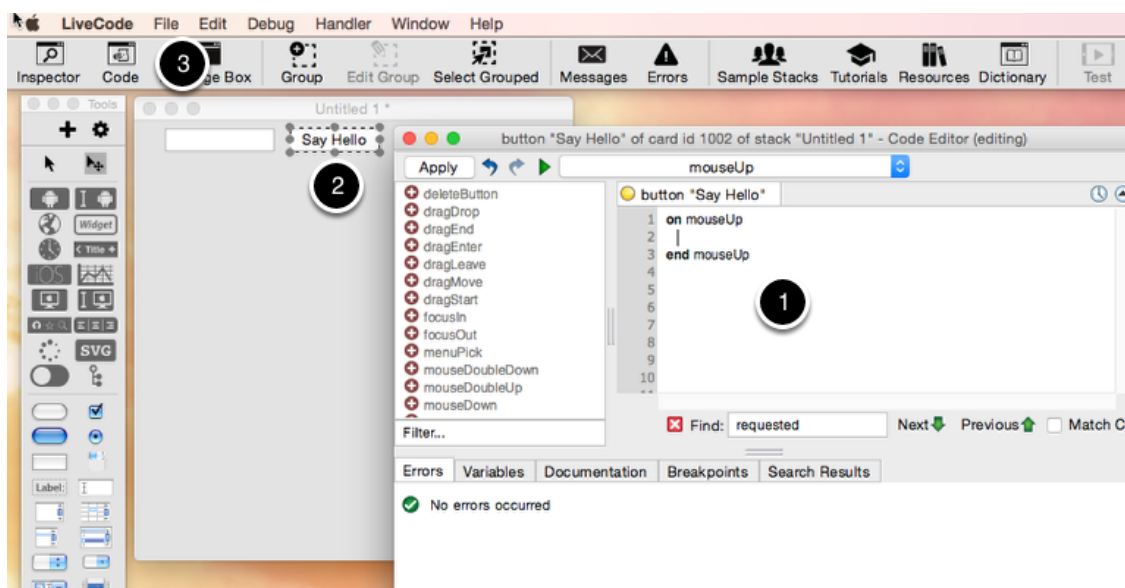
If we want our application to do anything we need to start adding some LiveCode. Once an object has been created and assigned properties its behaviour can be specified by giving it a script.

Within LiveCode all scripts are written in the script editor(1) The easiest way to view this is to select the object you want to edit(2) and then click the code button on the menubar(3). Alternatively the object can be right-clicked and the **Edit Script** option can be chosen from the menu which appears.

## Adding scripts to controls(2)
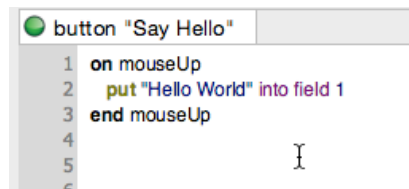
Scripts in LiveCode send and handle messages which control action. Some messages, such as mouseUp, are sent automatically by the system while other, user messages, are sent by making what is equivalent to a method call. The messages an object sends and how it handles them specify how it reacts to events and generally behaves.

In this case we want something to happen when the button is clicked. The mouseUp message is sent automatically when the mouse button is clicked and released so we add the script for whatever action we want to take to the mouseUp handler. Here we want to say hello so our script is
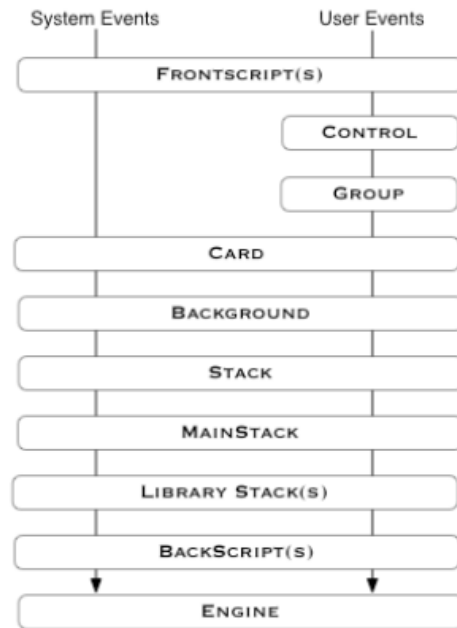
```
on mouseUp
        put "Hello World" into field 1
end mouseUp
```

Testing our script
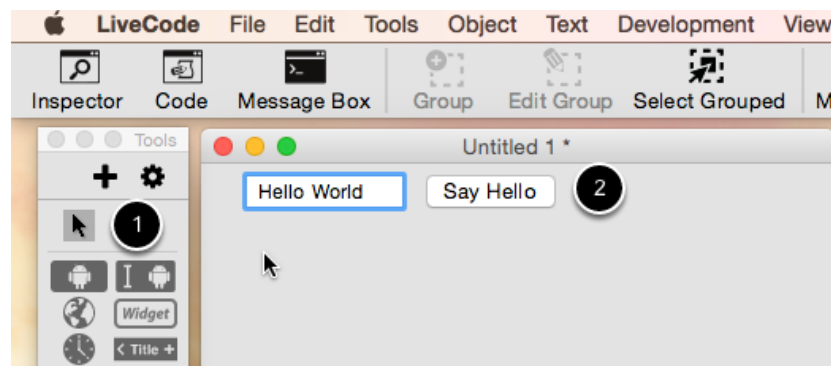
To test our script and see what happens we need to switch to **Run** mode(1) and click on the button(2) ... and Hello World appears in the field.

Messages and the Message Path

When a message is sent to an object, it is often handled directly by a message handler in that object. However if no handler is present, the message will continue along a path until it finds a message handler that can respond to it. This makes it possible to group similar functionality together at different levels within your application. This behavior applies both to event messages sent as a result of a user action, and custom messages sent by script. It is therefore possible to write libraries of common functions.

The object hierarchy is closely related to the *path* that a message travels on. In most cases, when an object passes a message on, the message goes to the object's owner in the object hierarchy.

For a more detailed explanation of the Message Path see the lesson How to call a function or command in another object? (http://lessons.livecode.com/m/4071/l/11787-how-to-call-a-function-or-command-in-another-object)

The Project Browser
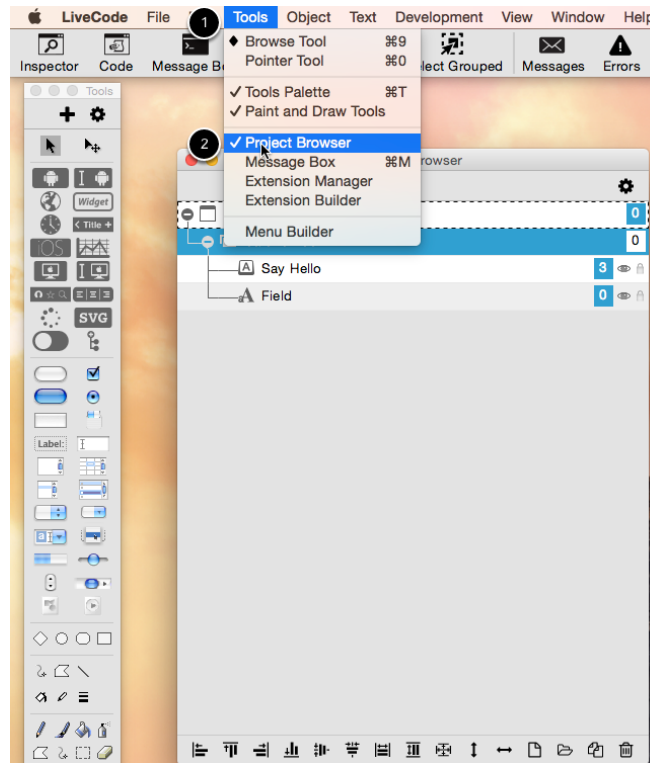
The Project Browser is a useful tool to see the structure of your stack and the controls that you have added.

The Project Browser contains a list of all open stacks, the cards in each stack, and the controls on each card. It allows you to navigate to any card, open or close a stack, select, open the property Inspector for, or edit the script of any object.

You can access the Project Browser by going to the Tools Menu(1) and choosing Project Browser(2)

10   Comments

drex   Tuesday Jun 07 2011 at 05:24 AM

I have just started using livecode and I am impressed by, but I wanted to know if there was any kind of if then statement, as well as a start command. I want to make a program that starts a website depending on what address you type in

Elanor Buchanan   Monday Jul 25 2011 at 07:33 AM

Hi drex

Yes there is an "if" statement. Just look up "if" in the dictionary for more information.

You can use the "launch url" command to open a browser with a given url. Again you can find more information on this in the dictionary.

I hope that helps.

Elanor

Andrea Friday May 25 2012 at 01:11 PM

May i create gui controls in source code?
For example i would like to create an empty app and design the graphic interface from a text file...

Hanson Schmidt-Cornelius Monday May 28 2012 at 04:48 AM

Hi Andrea,

yes, you can create the graphics interface from script. You could also load a text file in that had some kind of graphics syntax that you could parse and execute.

Have a look at the following lesson. This describes how to create the 8-Puzzle game, only using LiveCode to create the graphical user interface:

http://lessons.runrev.com/s/lessons/m/4071/l/15454-how-do-i-create-an-8-puzzle-game
(http://lessons.runrev.com/s/lessons/m/4071/l/15454-how-do-i-create-an-8-puzzle-game)

Kind Regards,

Hanson

Andrea Monday May 28 2012 at 01:45 PM

Many thanks for your reply!
With the CREATE command it seems I can create controls like:
"A control is any object that can be placed on a card: a button, field, scrollbar, image, graphic, player, EPS object, or group."
But there are many controls: grid, listbox, ecc. What about these other type of controls? Is there a way to create native os controls via code?

Hanson Schmidt-Cornelius Tuesday May 29 2012 at 05:50 AM

Hi Andrea,

the lesson I pointed you to shows you how to create simple buttons as follows:
new button "New Button"

You can then change the style of a button using something like:
set the style of button "New Button" to checkBox

You can create a list box like this:

new field "New Field"

put "line 1" & return & "line 2" & return & "line 3" & return & "line 4" into field "New Field"

set the vScrollbar of field "New Field" to true

The following lesson shows you how to create a data grid from script:

http://lessons.runrev.com/s/lessons/m/datagrid/l/36470-how-do-i-create-a-datagrid-dynamically-from-tab-delimited-data

(http://lessons.runrev.com/s/lessons/m/datagrid/l/36470-how-do-i-create-a-datagrid-dynamically-from-tab-delimited-data)

For native iOS controls, have a look at the dictionary entry:

mobileControlSet

Kind Regards,

Hanson

Jim Chamberlain Sunday Sep 09 2012 at 08:20 PM

How can I get information from a check box and what type is it?
thanks,
Jim

Hanson Schmidt-Cornelius Tuesday Sep 11 2012 at 06:43 AM

Hi Jim,

you can get the information of a check box by testing for its hilite. The result is either true or false. Try the following code from the card: put the hilite of button "yourButtonName"
If you are in the button script itself, you can do the following: put the hilite of me

Kind Regards,

Hanson

V Saturday Aug 17 2013 at 01:03 AM

Hi,
I'm new to LiveCode, and I was wondering how to make screen transitions.
Thanks,
V

Hanson Schmidt-Cornelius Tuesday Aug 20 2013 at 07:36 AM

Hi V,

you can make transitions to other cards. Have a look that the following lesson for basic navigation: http://lessons.runrev.com/s/lessons/m/4071/l/27527 (http://lessons.runrev.com/s/lessons/m/4071/l/27527)

In the dictionary also have a look at the command: visual effect

Kind Regards,

Hanson