

LiveCode 8.0.0-dp-2 Release Notes

Table of contents

Overview

- I don't want to build extensions. What's in it for me?

- LiveCode Script vs LiveCode Builder

- Warning

- IDE

Known issues

Platform support

- Windows

- Linux

- Mac

Setup

- Installation

- Uninstallation

Reporting installer issues

Activation

Multi-user and network install support (4.5.3)

Command-line installation

Command-line activation

Engine changes

- Image metadata

- LiveCode Builder

 - LiveCode Builder Language

 - Extensions

 - Getting Started

- Feature: Popup Widgets

 - New Syntax:

- 'sort cards of background ...' crashes

- Add revBrowser error callback messages.

 - Callback messages:

- New iOS status bar style

- [[In App Purchase]] Calling mobileStoreRestorePurchases when there are no previous purchases to restore

- Specific bug fixes (8.0.0-dp-2)

- Specific bug fixes (8.0.0-dp-1)

IDE changes

- IDE stackfiles named with version.

- Specific bug fixes (8.0.0-dp-1)

LiveCode Builder changes

- LiveCode Builder Tools

 - Compiler generates an error if integer literal too big

 - lc-compile

 - Warnings

 - Bugs fixed

- LiveCode Builder Language

- Replace concept of 'undefined' with 'nothing'

- Syntax

- Change to handler return type syntax.

- Identifiers

- Case-Sensitivity

- Bugs fixed

- LiveCode Builder Host Library

- Ability to access a widget's effective font

- Ability to display a popup menu

- Detecting successive clicks

- Widget Printing

- Bugs fixed

- LiveCode Tools

- Bugs fixed

- LiveCode Builder Standard Library

- Sequence operations

- Mathematical functions

- Dictionary changes

- Previous Release Notes

Overview

LiveCode 8.0 is the most exciting release in the history of the technology. It provides a simple way to extend the functionality or control set of LiveCode.

Our focus in LiveCode 8.0 is extensibility. You can now build and share widgets (custom controls) and libraries that are treated by LiveCode as engine level elements.

LiveCode 8.0 can be thought of as a version 7.0 with a new module allowing extensions to be plugged into the engine. As a result, 8.0 should be as functional and stable as LiveCode 7.0.

I don't want to build extensions. What's in it for me?

Many love LiveCode because of the productivity benefits and don't have time to build extensions. If that is the case just kick back and start using LiveCode 8 and keep an eye on the extensions portal. You can start using new controls and libraries as they are built by other community members.

LiveCode Script vs LiveCode Builder

To make it possible to create extensions and plug them into the LiveCode engine we've created a new flavour of our language called . LiveCode Builder looks a lot like LiveCode Script so should feel familiar for any seasoned LiveCode developer. There is lots of new syntax which exposes parts of the LiveCode engine that were only previously available to those who were skilled c/c++ developers.

LiveCode Builder is a new language and is therefore highly experimental and should be considered an early prototype. It will take some getting used to but we know you'll love it once you see how powerful it is. The best way to get started is to read the "Extending LiveCode" guide which can be found in the dictionary under the "Guide" tab.

Warning

It is important to stress that **no aspect of this release should be considered final. Every piece of syntax in LiveCode Builder is subject to change**

Known issues

- The installer will currently fail if you run it from a network share on Windows. Please copy the installer to a local disk before launching on this platform.
- The new property inspector lacks some properties present in the old property inspector
- The supplied widgets are examples and lack features and general robustness
- The extension builder plugin "Test" feature fails if the widget being tested is already installed - uninstalling the widget and restarting the IDE should help
- All installed widgets are built into any standalones produced

Platform support

The engine supports a variety of operating systems and versions. This section describes the platforms that we ensure the engine runs on without issue (although in some cases with reduced functionality).

Windows

The engine supports the following Windows OSes:

- Windows XP SP2 and above
- Windows Server 2003
- Windows Vista SP1 and above (both 32-bit and 64-bit)
- Windows 7 (both 32-bit and 64-bit)
- Windows Server 2008
- Windows 8.x (Desktop)

Note: On 64-bit platforms the engine still runs as a 32-bit application through the WoW layer.

Linux

The linux engine requires the following:

- Supported architectures:

32-bit or 64-bit Intel/AMD or compatible processor

32-bit ARMv6 with hardware floating-point (e.g. RaspberryPi)

- Common requirements for GUI functionality:

GTK/GDK/Glib 2.24 or later

Pango with Xft support

(optional) esd - required for audio output

(optional) mplayer - required for media player functionality

(optional) lcms - required for color profile support in images

(optional) gksu - required for privilege elevation support

- Requirements for 32-bit Intel/AMD:

glibc 2.3.6 or later

- Requirements for 64-bit Intel/AMD:

glibc 2.15 or later

- Requirements for ARMv6:

glibc 2.7 or later

Note: The GUI requirements are also required by Firefox and Chrome, so if your Linux distribution runs one of those, it will run the engine.

Note: If the optional requirements are not present then the engine will still run but the specified features will be disabled.

Note: It may be possible to compile and run LiveCode Community on other architectures but this is not officially supported.

Mac

The Mac engine supports:

- 10.6.x (Snow Leopard) on Intel
- 10.7.x (Lion) on Intel

- *10.8.x (Mountain Lion) on Intel*
- *10.9.x (Mavericks) on Intel*

Note: The engine runs as a 32-bit application regardless of the capabilities of the underlying processor.

Setup

Installation

Each distinct version has its own complete folder – multiple versions will no longer install side-by-side: on Windows (and Linux), each distinct version will gain its own start menu (application menu) entry; on Mac, each distinct version will have its own app bundle.

The default location for the install on the different platforms when installing for 'all users' are:

- Windows: <x86 program files folder>/RunRev/ LiveCode 8.0.0-dp-2
- Linux: /opt/runrev/livecode-8.0.0-dp-2
- Mac: /Applications/ LiveCode 8.0.0-dp-2.app

The default location for the install on the different platforms when installing for 'this user' are:

- Windows: <user roaming app data folder>/RunRev/Components/LiveCode 8.0.0-dp-2
- Linux: ~/.runrev/components/livecode-8.0.0-dp-2
- Mac: ~/Applications/ LiveCode 8.0.0-dp-2.app

Note: *If your linux distribution does not have the necessary support for authentication (gksu) then the installer will run without admin privileges so you will have to manually run it from an admin account to install into a privileged location.*

Uninstallation

On Windows, the installer hooks into the standard Windows uninstall mechanism. This is accessible from the appropriate pane in the control panel.

On Mac, simply drag the app bundle to the Trash.

On Linux, the situation is currently less than ideal:

- open a terminal
- `cd` to the folder containing your rev install. e.g.

```
cd /opt/runrev/livecode-8.0.0-dp-2
```

- execute the `.setup.x86` file. i.e.

```
./setup.x86
```

- follow the on-screen instructions.

Reporting installer issues

If you find that the installer fails to work for you then please file a bug report in the RQCC or email support@runrev.com so we can look into the problem.

In the case of failed install it is vitally important that you include the following information:

- Your platform and operating system version
- The location of your home/user folder
- The type of user account you are using (guest, restricted, admin etc.)
- The installer log file located as follows:
 - **Windows 2000/XP:** <documents and settings folder>/<user>/Local Settings/

- **Windows Vista/7:** <users folder>/<user>/AppData/Local/RunRev/Logs
- **Linux:** <home>/runrev/logs
- **Mac:** <home>/Library/Application Support/Logs/RunRev

Activation

The licensing system ties your product licenses to a customer account system, meaning that you no longer have to worry about finding a license key after installing a new copy of LiveCode. Instead, you simply have to enter your email address and password that has been registered with our customer account system and your license key will be retrieved automatically.

Alternatively it is possible to activate the product via the use of a specially encrypted license file. These will be available for download from the customer center after logging into your account. This method will allow the product to be installed on machines that do not have access to the internet.

Multi-user and network install support (4.5.3)

In order to better support institutions needing to both deploy the IDE to many machines and to license them for all users on a given machine, a number of facilities have been added which are accessible by using the command-line.

Note: *These features are intended for use by IT administrators for the purposes of deploying LiveCode in multi-user situations. They are not supported for general use.*

Command-line installation

It is possible to invoke the installer from the command-line on both Mac and Windows. When invoked in this fashion, no GUI will be displayed, configuration being supplied by arguments passed to the installer.

On both platforms, the command is of the following form:

<exe> install noui *options*

Here *options* is optional and consists of one or more of the following:

-allusers	Install the IDE for all users. If not specified, the install will be done for the current user only.
-	
desktopshortcut	Place a shortcut on the Desktop (Windows-only)
-startmenu	Place shortcuts in the Start Menu (Windows-only)
-location	The location to install into. If not specified, the location defaults to those described in the <i>Layout</i> section above.
<i>location</i>	
-log <i>logfile</i>	A file to place a log of all actions in. If not specified, no log is generated.

Note that the command-line variant of the installer does not do any authentication. Thus, if you wish to install to an admin-only location you will need to be running as administrator before executing the command. As the installer is actually a GUI application, it needs to be run slightly differently from other command-line programs.

In what follows <installerexe> should be replaced with the path of the installer executable or app (inside the DMG) that has been downloaded.

On Windows, you need to do:

```
start /wait <installerexe> install noui options
```

On Mac, you need to do:

```
"<installerexe>/Contents/MacOS/installer" install noui options
```

On both platforms, the result of the installation will be written to the console.

Command-line activation

In a similar vein to installation, it is possible to activate an installation of LiveCode for all-users of that machine by using the command-line. When invoked in this fashion, no GUI will be displayed, activation being controlled by any arguments passed.

On both platforms, the command is of the form:

```
<exe> activate -file license -passphrase phrase
```

This command will load the manual activation file from *license*, decrypt it using the given *passphrase* and then install a license file for all users of the computer. Manual activation files can be downloaded from the 'My Products' section of the RunRev customer accounts area.

This action can be undone using the following command:

```
<exe> deactivate
```

Again, as the LiveCode executable is actually a GUI application it needs to be run slightly differently from other command-line programs.

In what follows <livecodeexe> should be replaced with the path to the installed LiveCode executable or app that has been previously installed.

On Windows, you need to do:

```
start /wait <livecodeexe> activate -file license -passphrase phrase
```

```
start /wait <livecodeexe> deactivate
```

On Mac, you need to do:

```
"<livecodeexe>/Contents/MacOS/LiveCode" activate -file license -passphrase phrase
```

```
"<livecodeexe>/Contents/MacOS/LiveCode" deactivate
```

On both platforms, the result of the activation will be written to the console.

Engine changes

Image metadata (8.0.0-dp-2)

A new read only image property has been added to access the metadata in the image file. The returned array is in the same format as that used for the export command. If no metadata is found then the property returns empty rather than an array with empty elements. Currently the only metadata key that is implemented is `density` which can be used to determine pixel density in pixels per inch. Metadata is currently only parsed from JPEG and PNG file formats.

For example:

```
put the metadata of image 1 into metadataArray
```

```
set the width of image 1 to the width of image 1 div (metadataArray["density"] / 72)
```

```
set the height of image 1 to the height of image 1 div (metadataArray["density"] / 72)
```

LiveCode Builder (8.0.0-dp-2)

LiveCode Builder Language

LiveCode Builder is a variant of the current LiveCode scripting language (LiveCode Script) which has been designed for 'systems' building. It is statically compiled with optional static typing and direct foreign code interconnect (allowing easy access to APIs written in other languages). The compiled bytecode can then be packaged together with any required resources (icons, documentation, images, etc) into a .lcb extension package.

Unlike most languages, LiveCode Builder (LCB) has been designed around the idea of extensible syntax. Indeed, the core language is very small - comprising declarations and control structures - with the majority of the language syntax and functionality being defined in modules.

Note: It is an eventual aim that control structures will also be extensible, however this is not the case in the current incarnation).

The syntax will be familiar to anyone who has coded with LiveCode Script, however LiveCode Builder is a great deal more strict - the reason being it is intended that it will eventually be compilable to machine code with the performance and efficiency you'd expect from any 'traditional' programming language. Indeed, over time we hope to move the majority of implementation of the whole LiveCode system over to being written in LiveCode Builder.

Note: One of the principal differences is that type conversion is strict - there is no automatic conversion between different types such as between number and string. Such conversion must be explicitly specified using syntax (currently this is done using syntax like *... parsed as number* and *... formatted as string*).

Extensions

There are two types of extensions which can be written in LCB: widgets and libraries. All installed extensions appear in the new Extension Manager stack, which can be opened from the Tools menu.

An LCB library is a new way of adding functions to the LiveCode message path. Public handlers in loaded LCB libraries are available to call from LiveCode Script.

A widget is a new type of custom control which, once compiled and packaged, can be loaded into the IDE.

Using the widget is no different from any of the classic LiveCode controls you've been used to. Simply drag it onto a stack and start interacting with it as you would any another control.

You can reference the widget in script as a control:

```
set the name of the last control to "clock"
```

Or more specifically as a widget:

```
set the tooltip of widget 1 to "This is my nice new clock widget"
```

Getting Started

To get started with LiveCode Builder, click on the "Dictionary" icon in the IDE toolbar, select the "Guide" tab and then "Extending LiveCode" from the drop-down menu. This will show you the user-guide on getting started with writing widgets and libraries in LCB. Alternatively, you can start by looking at some of the extensions shipped with LiveCode 8 - the source and other resources for these are located in the "extensions" sub-folder of your LiveCode installation directory (source files are named

Feature: Popup Widgets (8.0.0-dp-2)

Added the ability to use widgets within popup dialog windows.

New Syntax:

- popup widget <Kind> at <Position> [with properties <Properties>]
- Launch the named widget as a popup. The popup can return a value in the result.
- currently popped up
- test if this widget is part of a popup
- close popup [returning <Result>]
- set the result of the calling popup statement to <Result>

'sort cards of background ...' crashes (8.0.0-dp-2)

You cannot 'sort cards of background' so this now throws an error, rather than crashing.

Add revBrowser error callback messages. (8.0.0-dp-1)

Added two new callbacks sent by revBrowser.

Note that these messages are only sent from browsers opened with revBrowserOpenCEF.

Callback messages:

- browserDocumentFailed
- sent to the current card when the browser has encountered an error while loading a URL.
- browserDocumentFailedFrame
- sent to the current card when the browser has encountered an error while loading a URL into a frame.

New iOS status bar style (8.0.0-dp-1)

What has changed?

The way the status bar is displayed changed in iOS 7. Previously, if the status bar was visible,

the app view was shifted down by height of the status bar (20 pixels). From iOS 7 and above, the app view is given these 20 pixels of extra height, and the status bar is displayed on top of the app view (i.e there is an overlapping). This is a feature added by Apple, to give developers control over what appears behind the content of the status bar.

Some users reported this change of behaviour as a bug, since it changed the way their existing stacks were displayed in iOS 7 and above. A quick fix to this would be updating the engine to move the app view down 20 pixels, if the status bar is opaque. However, this fix would not guarantee backwards compatibility, since some users may have already adjusted for iOS 7 status bar behaviour, and have modified their code to work with those changes.

So we decided to add a new **solid** status bar style, which is opaque and automatically shifts down the view content by 20 pixels. So the difference between "opaque" and "solid" status bar styles is the following:

| pre-iOS 7 | iOS 7+

-----|-----|-----

opaque | move the stack below the status bar | the status bar will be over the top
solid | move the stack below the status bar | move the stack below the status bar

[[In App Purchase]] Calling mobileStoreRestorePurchases when there are no previous purchases to restore (8.0.0-dp-1)

What has changed

Previously, if mobileStoreRestorePurchases was called and no previous purchases were made with that user account, nothing happened.

Now, a purchaseStateUpdate message is sent with state=restored and productID=""

Specific bug fixes (8.0.0-dp-2)

(bug fixes specific to the current build are highlighted in bold, reverted bug fixes are stricken through)

- 15140 **String list parsing for properties incorrect**
- 15062 **Standalones saved in a folder with Unicode char will not launch on Mac**
- 15056 **Read from file for (x | x chars | x bytes) returns empty**
- 15045 **Parse issue with nested char chunk expression**
- 15037 **Simulators listed in Preference > Mobile Support are not in the Development > Test Target list**
- 15034 **import snapshot does not work correctly on 64-bit Linux.**
- 15011 **Crash when sorting non-shared text field in a group**
- 14996 **LCB-Canvas: polyline path**
- 14965 **setting cursor to id of non-existing image crashes LC 7.0.4 rc1**
- 14961 **Gradient - Quality set to "good" makes LC crash**
- 14960 **LC8 DP1 crash while importing JPG image**
- 14946 **More Room For Prompt/Message in mobile busy indicator**
- 14885 **'sort cards of background ...' crashes**
- 14866 **Can't Use XCode Version 6.2 (6C131e)**
- 14858 **[In-app purchase] Calling mobileStoreProductProperty(prodid**
- 14855 **mobileSensorReading(sensor_name**
- 14833 **Format strings of the form "%0s" do not work correctly.**

- 14816 iOS fullscreen video controller not visible when tapping on movie rect
- 14814 print card€ into pageRect fails in LC7.0.3
- 14779 [in-app purchase] Android: Receive a purchaseStateUpdate message with state=complete after calling mobileStoreConfirmPurchase
- 14752 Quitting LiveCode with revBrowserCEF window open causes a hang
- 14712 Save As converts names to lower-case on Linux
- 14689 Option menu crashes app in Simulator 8.0
- 14661 LiveCode 7 IDE Doesn't Use Standard Windows Font
- 14612 read from file COM1: --- port doesn't work ---
- 14611 answer file/folder .. with .. does not work anymore with backslash delimiter
- 14489 MobilePick Fails on Android 5 every other load
- 14408 Using beep while audio is playing on mobile
- 14326 revZip externals does not work on iOS
- 14160 LiveCode 7.0 silently crashes if the Print Spooler is turned off
- 14139 ChartsEngine 1.2.1 can't be registered with LiveCode 7
- 13969 Wrong Splash screen on iPhone 6+
- 8924 Script Editor incorrectly highlights block comments as if they nest.

Specific bug fixes (8.0.0-dp-1)

- 14851 Popup won't stop displaying when displayed in mouseDown of button widget
- 14665 fatal exception - Service not registered: com.runrev.android.billing.google.labHelper
- 14658 Recursive array ops fail debug mode mutability assertion
- 14604 [LC 7] iOS mobile player 'loadstate' property never contains 'playthrough'
- 14602 URLEncode crashes LiveCode
- 14599 LCB: Text sort is inconsistent with string comparison
- 14588 Intermittent crash on MCPlatformWindowMaskRelease()
- 14587 LC 7 Server buffers output
- 14538 bool formatted as string does not work
- 14536 Return parameters require a local variable to be defined first
- 14531 EOF is returned when reading exactly the number of chars in a file
- 14525 video controller not responding
- 14521 IDE crashes when viewing with Inspector
- 14515 put URL into URL"binFile:eXample.mp3" crash on Android
- 14504 Strings which have been 'unnativized' by the engine lose the CanBeNative flag when copied
- 14483 fiddling with alphaData crashes/hangs LC 7.0.2
- 14480 Typo in textEncode dictionary entry
- 14477 Trying to get \$_FILES array fails since server version 7.
- 14467 Unable to get line Attributes of Char Chunks in LC 7
- 14463 Second invocation of play command causes crash on iOS
- 14462 Crash related to shell command on Linux
- 14461 [[In-app purchase]] mobileStorePurchasedProducts returns empty on iOS
- 14457 Segment-based URIs yield an error since server version 7
- 14456 Google Play Store - No Information on Product Details
- 14450 Google Play Store - Extra error message when an in-app purchase fails
- 14441 Shell command takes much longer to run in LiveCode Server 7 than in 6

- 14439 Minus sign ignored when assigning value to variable at declaration
- 14437 Clearing a Player filename causes LC to crash.
- 14402 [[In App Purchase]] Calling mobileStoreRestorePurchases when there are no previous purchases to restore
- 14383 Android standalones can not open a new file for update on the device
- 14140 Location of modal stacks incorrect on OSX
- 14011 Printing is inconsistent
- 13586 LC and standalones use wrong version icon in Ubuntu Launcher
- 11544 iOS 7 Status Bar Hidden

IDE changes

IDE stackfiles named with version. (8.0.0-dp-1)

When a binary stackfile is rewritten in the IDE for a new version, it should have a (major) version in the filename to prevent unwanted IDE merging between versions. This can also be used to ensure incompatible stacks are not loaded if present - the IDE will only load stacks with a version less than or equal to its version.

For example, from 8.0 onwards, revTools has filename /Toolset/revTools.8.rev.

Specific bug fixes (8.0.0-dp-1)

- 14627
- 14400 App requests notification permission when it isn't configured to use them
- 14373 Linux icon not checked when building for Linux64 or Linux ARM only
- 14348 Database drivers missing when building Linux 64-bit standalones
- 13475 in the openstack handler dispatching a mouseUp to a btn does not work correctly
- 13447 Project Browser control layer display
- 13417 IDE systemVersion comparison no longer works with Yosemite
- 13398 Sample - Book Library.livecode edit and delete features broken
- 13362 Script editor opens revmenubar script when no other stack is open
- 13343 Cannot install Android standalone on some devices
- 13215 Can't type in output field of message box
- 13191 FIX: flip graphic horizontally and vertically for complex graphics
- 13159 Palettes not observing decorations under certain circumstances
- 12880 File->Exit should be File->Quit
- 11755 flip graphic gives erroneous results with complex graphics

LiveCode Builder changes

LiveCode Builder Tools

Compiler generates an error if integer literal too big

The compiler will generate an error if an integer literal is too big to fit into the (current) unsigned 32-bit integer representation.

lc-compile

Warnings

- A new warning has been added for identifiers that may conflict with syntax keywords.
- `metadata` definitions that occur before module imports no longer

trigger a warning.

Bugs fixed

[14893] Emit warnings for identifiers that may cause problems.

[14939] Compiler truncates integer literals if too big.

[14950] Integer literal pattern is too general.

[14956] Do not warn for `metadata before use`.

[15029] Ensure that `exit repeat` works correctly in nested `repeat up to / down to /` for each loops.

LiveCode Builder Language

Replace concept of 'undefined' with 'nothing'

- The use of the keyword 'undefined' is now deprecated, 'nothing' should be used instead.
 - Use 'returns nothing' to indicate a handler which returns no value.
 - Use 'nothing' to indicate no value when manipulating optionally type variables
- The 'is defined', 'is undefined', 'is not defined', 'is not undefined' syntax is now deprecated, 'is' and 'is not' should be used with 'nothing' instead
 - Use `<expr> is nothing` and `<expr> is not nothing` to test whether an expression has a value or not
 - The phrase `<left> is <right>` will now return true if `<left>` and `<right>` are both nothing
 - The phrase `<left> is not <right>` will now return true if one of `<left>` or `<right>` are nothing (but not both).

Syntax

- Syntax keywords are no longer permitted to match `[A-Z0-9_.]`.

- `metadata` definitions may now occur anywhere a module's top-level

context.

Change to handler return type syntax.

- The syntax for declaring the return type for a handler, or handler type has been changed to [`'returns' 'nothing' | 'returns' <Type>]`.
- The previous syntax `as <Type>` or `as undefined` will continue to be supported until dp-4 at which point it will be removed along with the `undefined` type keyword.
- The compiler will emit warnings for the use of the deprecated syntax.

Identifiers

- Identifiers are now expected to match `[A-Z0-9_.]`.

Case-Sensitivity

- All identifiers are now case-insensitive - i.e. a handler `Main` can be called as `mAin`, `MAIN` and `main`.

Bugs fixed

[14906] Change `as <Type>` to `'returns nothing'` or `'returns <Type>'` in handler return type definitions.
 [14933] Make identifiers case-insensitive.

LiveCode Builder Host Library

Ability to access a widget's effective font

- The `textFont`, `textSize` and `textStyle` properties have been reserved to the host.
- New syntax `my font` has been added which returns a `Canvas.Font` matching the current effective values of the text properties that have been set on the widget.

Ability to display a popup menu

- New syntax has been added to popup a menu constructed from a provided menu text.
 - `popup menu <MenuText> at <Point>`

Detecting successive clicks

- The `OnClick` event is sent every time a `mouseDown/mouseUp` sequence is detected by the engine on a widget.
- Use `'the click count'` syntax to fetch the number of successive clicks which happened close together and within a certain time of each other.

Widget Printing

- Widgets now print along with other controls.
 - Widgets will be rasterized at screen resolution and then printed as an image.
 - Higher-fidelity printing of widgets will be implemented at a later date.

Bugs fixed

- [14541] Widgets should be able to popup system menus
- [14805] LCB-Canvas: close path on mPath example is incorrect
- [14898] Add access to widget's effective font.
- [14964] Crash when calling a handler with 'any' type parameter form LCS.
- [14997] image with file accepts url prefix
- [15005] Logging a list fails if its element types are not strings or booleans
- [15012] Fix OnClick and provide a way to get the number of successive clicks.
- [15035] Svg path loses data in roundtrip
- [15060] Printing a card containing a widget causes a crash.
- [15128] OnMouseMove doesn't fire if there is more than one widget on a card.
- [15138] Map elements inside lists to arrays correctly.

LiveCode Tools

Bugs fixed

- [14938] The compiler now checks a property get handler does not return nothing.

LiveCode Builder Standard Library

Sequence operations

- New syntax has been added for searching partial contents of sequence types (`List`, `String` and `Data`) based on the `offset` operation.
 - the offset of `<Needle>` before `<Position>` in `<Haystack>`
 - the offset of `<Needle>` after `<Position>` in `<Haystack>`
 - Equivalent syntax has been added for the `index` operation.

Mathematical functions

- Several mathematical functions now throw "domain errors" when applied to values that the function is not defined for, including

`log10()`, `ln()`, `asin()` and `acos()`, and `x ^ y`.

Bugs fixed

- [14678] Throw error on domain error in `log10()` and `ln()`
- [14679] Throw error on domain error in `pow()`
- [14681] Throw error on domain error in `asin()` and `acos()`
- [14846] Fixed and reinstated "offset before" and "offset after"
- [14889] Throw error when "converted from base _" fails

Dictionary changes

- The entry for **resolve image** (*command*) has been updated.
- The entry for **HTMLText** (*property*) has been updated.
- The entry for **activatePalettes** (*property*) has been updated.
- The entry for **innerGlow** (*property*) has been updated.
- The entry for **innerGlow** (*property*) has been updated.
- The entry for **password** (*property*) has been updated.
- The entry for **serialControlString** (*property*) has been updated.
- The entry for **sslCertificates** (*property*) has been updated.
- The entry for **sslCertificates** (*property*) has been updated.
- The entry for **stackFiles** (*property*) has been updated.

Previous Release Notes

7.0.3 Release Notes	http://downloads.livecode.com/livecode/7_0_3/LiveCodeNotes-7_0_3.pdf
7.0.1 Release Notes	http://downloads.livecode.com/livecode/7_0_1/LiveCodeNotes-7_0_1.pdf
7.0.0 Release Notes	http://downloads.livecode.com/livecode/7_0_0/LiveCodeNotes-7_0_0.pdf
6.7.2 Release Notes	http://downloads.livecode.com/livecode/6_7_2/LiveCodeNotes-6_7_2.pdf
6.7.1 Release Notes	http://downloads.livecode.com/livecode/6_7_1/LiveCodeNotes-6_7_1.pdf
6.7.0 Release Notes	http://downloads.livecode.com/livecode/6_7_0/LiveCodeNotes-6_7_0.pdf
6.6.2 Release Notes	http://downloads.livecode.com/livecode/6_6_2/LiveCodeNotes-6_6_2.pdf
6.6.1 Release Notes	http://downloads.livecode.com/livecode/6_6_1/LiveCodeNotes-6_6_1.pdf
6.6.0 Release Notes	http://downloads.livecode.com/livecode/6_6_0/LiveCodeNotes-6_6_0.pdf
6.5.2 Release Notes	http://downloads.livecode.com/livecode/6_5_2/LiveCodeNotes-6_5_2.pdf
6.5.1 Release Notes	http://downloads.livecode.com/livecode/6_5_1/LiveCodeNotes-6_5_1.pdf
6.5.0 Release Notes	http://downloads.livecode.com/livecode/6_5_0/LiveCodeNotes-6_5_0.pdf
6.1.3 Release Notes	http://downloads.livecode.com/livecode/6_1_3/LiveCodeNotes-6_1_3.pdf
6.1.2 Release Notes	http://downloads.livecode.com/livecode/6_1_2/LiveCodeNotes-6_1_2.pdf
6.1.1 Release Notes	http://downloads.livecode.com/livecode/6_1_1/LiveCodeNotes-6_1_1.pdf
6.1.0 Release Notes	http://downloads.livecode.com/livecode/6_1_0/LiveCodeNotes-6_1_0.pdf
6.0.2 Release Notes	http://downloads.livecode.com/livecode/6_0_2/LiveCodeNotes-6_0_2.pdf
6.0.1 Release Notes	http://downloads.livecode.com/livecode/6_0_1/LiveCodeNotes-6_0_1.pdf
6.0.0 Release Notes	http://downloads.livecode.com/livecode/6_0_0/LiveCodeNotes-6_0_0.pdf