LiveCode 9.0.1 Release Notes

Overview

LiveCode 9.0 enables access to libraries and platform APIs written in many other languages thanks to the community-funded 'Infinite LiveCode' project.

This includes a greatly improved LiveCode Builder virtual machine.

LiveCode 9.0 contains many additional improvements to support LiveCode app developers, including:

- A new "spinner" widget
- OAuth2 authentication library for use with web APIs (e.g. Facebook, Google and GitHub)
- A command argument parser library for building command-line standalones
- Updates and performance improvements for existing widgets

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 browser widget does not work on 32-bit Linux.
- 64-bit standalones for Mac OS X do not have support for audio recording.

Breaking changes

Standalone Building

The standalone builder has always needed to close the stacks it builds for reasons pretty deeply ingrained in the code. However this causes a few problems, for example:

- values in script locals become empty
- behaviors are broken when the parent script is on / in a stack which closes

As an attempt to improve this situation, the code that locks messages when closing and opening stacks for standalone builds has been removed. This means that where previously mainstacks would not receive openStack and closeStack messages during standalone build, they now do.

If this causes problems for your stack, you can exit from the handler if standalone building is in progress:

```
on closeStack
    if the mode of stack "revStandaloneProgress" > 0 then
        exit closesStack
    end if
end closeStack
```

LiveCode Builder

Exponentiation operator precedence

Prior to this release, exponentiation had lower precedence that unary minus. In order to write code that operates as expected in both this release and previous releases, please use parentheses where appropriate.

Using lc-compile tool in LiveCode 9:

```
-1^2 = -1
```

Using lc-compile tool in LiveCode 8:

```
-1^2 = 1
```

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

LiveCode supports the following versions of Windows:

- Windows 7 (both 32-bit and 64-bit)
- Windows Server 2008
- Windows 8.x (Desktop)

- Windows 10

Note: On 64-bit Windows installations, LiveCode runs as a 32-bit application through the WoW layer.

Linux

LiveCode supports the following Linux distributions, on 32-bit or 64-bit Intel/AMD or compatible processors:

- Ubuntu 14.04 and 16.04
- Fedora 23 & 24
- Debian 7 (Wheezy) and 8 (Jessie) [server]
- CentOS 7 [server]

LiveCode may also run on Linux installations which meet the following requirements:

- Required dependencies for core functionality:
    - glibc 2.13 or later
    - glib 2.0 or later
- Optional requirements for GUI functionality:
    - GTK/GDK 2.24 or later
    - Pango with Xft support
    - esd (optional, needed for audio output)
    - mplayer (optional, needed for media player functionality)
    - lcms (optional, required for color profile support in images)
    - gksu (optional, required for privilege elevation support)

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

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

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

Mac

The Mac engine supports:

- 10.9.x (Mavericks) on Intel
- 10.10.x (Yosemite) on Intel
- 10.11.x (El Capitan) on Intel
- 10.12.x (Sierra) on Intel
- 10.13.x (High Sierra) on Intel

iOS

iOS deployment is possible when running LiveCode IDE on a Mac, and provided Xcode is installed and has been set in LiveCode *Preferences* (in the *Mobile Support* pane).

Currently, the supported versions of Xcode are:

- Xcode 6.2 on MacOS X 10.9
- Xcode 6.2 and 7.2 on Mac OS X 10.10
- Xcode 8.2 on MacOS X 10.11
- Xcode 9.2 on MacOS 10.12 (Note: You need to upgrade to 10.12.6)
- Xcode 9.4 on MacOS 10.13 (Note: You need to upgrade to 10.13.2)

It is also possible to set other versions of Xcode, to allow testing on a wider range of iOS simulators. For instance, on MacOS 10.12 (Sierra), you can add *Xcode 6.2* in the *Mobile Support* preferences, to let you test your stack on the *iOS Simulator 8.2*.

We currently support deployment for the following versions of iOS:

- 8.2 [simulator]
- 9.2
- 10.2
- 11.2
- 11.4

Android

LiveCode allows you to save your stack as an Android application, and also to deploy it on an Android device or simulator from the IDE.

Android deployment is possible from Windows, Linux and Mac OSX.

The Android engine supports devices using ARMv7 or ARMv8 processors. It will run on the following versions of Android:

- 4.1-4.3 (Jelly Bean)
- 4.4 (KitKat)
- 5.0-5.1 (Lollipop)
- 6.0 (Marshmallow)
- 7.0 (Nougat)
- 8.0 (Oreo)

To enable deployment to Android devices, you need to download the Android SDK, and then use the 'Android SDK Manager' to install:

- the latest "Android SDK Tools"
- the latest "Android SDK Platform Tools"

You also need to install the Java Development Kit (JDK). On Linux, this usually packaged as "openjdk". LiveCode requires JDK version 1.6 or later.

Once you have set the path of your Android SDK in the "Mobile Support" section of the LiveCode IDE's preferences, you can deploy your stack to Android devices.

Some users have reported successful Android Watch deployment, but it is not officially supported.

HTML5

LiveCode applications can be deployed to run in a web browser, by running the LiveCode engine in JavaScript and using modern HTML5 JavaScript APIs.

HTML5 deployment does not require any additional development tools to be installed.

LiveCode HTML5 standalone applications are currently supported for running in recent versions of Mozilla Firefox, Google Chrome or Safari. For more information, please see the "HTML5 Deployment" guide in the LiveCode IDE.

Setup

Installation

Each version of LiveCode installs can be installed to its own, separate folder. This allow multiple versions of LiveCode to be installed side-by-side. On Windows (and Linux), each version of LiveCode has its own Start Menu (or application menu) entry. On Mac OS X, each version has its own app bundle.

On Mac OS X, install LiveCode by mounting the `.dmg` file and dragging the app bundle to the `Applications` folder (or any other suitable location).

For Windows and Linux, the default installation locations when installing for "All Users" are:

Platform	Path
Windows	`<x86 program files folder>/RunRev/LiveCode <version>`
Linux	`/opt/livecode/livecode-<version>`

The installations when installing for "This User" are:

Platform	Path
Windows	`<user roaming app data folder>/RunRev/Components/LiveCode <version>`
Linux	`~/.runrev/components/livecode-<version>`

Note: If installing for "All Users" on Linux, either the **gksu** tool must be available, or you must manually run the LiveCode installer executable as root (e.g. using **sudo** or **su**).

Uninstallation

On Windows, the installer hooks into the standard Windows uninstall mechanism. This is accessible from the "Add or Remove Programs" applet in the windows Control Panel.

On Mac OS X, drag the app bundle to the Trash.

On Linux, LiveCode can be removed using the `setup.x86` or `setup.x86_64` program located in

LiveCode's installation directory.

Reporting installer issues

If you find that the installer fails to work for you then please report it using the LiveCode Quality Control Centre or by emailing support@livecode.com.

Please include the following information in your report:

- Your platform and operating system version
- The location of your home or user folder
- The type of user account you are using (guest, restricted, admin etc.)
- The installer log file.

The installer log file can be located as follows:

Platform	Path
Windows 2000/XP	`<documents and settings folder>/<user>/Local Settings/`
Windows Vista/7	`<users folder>/<user>/AppData/Local/RunRev/Logs`
Linux	`<home>/.runrev/logs`

Activating LiveCode Indy or Business edition

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.

Command-line installation

It is possible to invoke the installer from the command-line on Linux and Windows. When doing command-line installation, no GUI will be displayed. The installation process is controlled by arguments passed to the installer.

Run the installer using a command in the form:

```
<installer> install noui [OPTION ...]
```

where `<installer>` should be replaced with the path of the installer executable or app (inside the DMG) that has been downloaded. The result of the installation operation will be written to the console.

The installer understands any of the following `OPTION`s:

Option	Description
`-allusers`	Install the IDE for "All Users". If not specified, LiveCode will be installed for the current user only.
`-desktopshortcut`	Place a shortcut on the Desktop (Windows-only)
`-startmenu`	Place shortcuts in the Start Menu (Windows-only)
`-location LOCATION`	The folder to install into. If not specified, the `LOCATION` defaults to those described in the "Installation" section above.
`-log LOGFILE`	The file to which to log installation actions. If not specified, no log is generated.

Note: the command-line installer does not do any authentication. When installing for "All Users", you will need to run the installer command as an administrator.

As the installer is actually a GUI application, it needs to be run slightly differently from other command-line programs.

On Windows, the command is:

```
start /wait <installer> install noui [OPTION ...]
```

Command-line uninstallation

It is possible to uninstall LiveCode from the command-line on Windows and Linux. When doing command-line uninstallation, no GUI will be displayed.

Run the uninstaller using a command of the form:

```
<uninstaller> uninstall noui
```

Where is *.setup.exe* on Windows, and *.setup.x86* on Linux. This executable, for both of the platforms, is located in the folder where LiveCode is installed.

The result of the uninstallation operation will be written to the console.

Note: the command-line uninstaller does not do any authentication. When removing a version of LiveCode installed for "All Users", you will need to run the uninstaller command as an administrator.

Command-line activation for LiveCode Indy or Business edition

It is possible to activate an installation of LiveCode for all users by using the command-line. When performing command-line activation, no GUI is displayed. Activation is controlled by passing command-line arguments to LiveCode.

Activate LiveCode using a command of the form:

```
<livecode> activate -file LICENSEFILE -passphrase SECRET
```

where `<livecode>` should be replaced with the path to the LiveCode executable or app that has been previously installed.

This loads license information from the manual activation file `LICENSEFILE`, decrypts it using the given `SECRET` passphrase, and installs a license file for all users of the computer. Manual activation files can be downloaded from the My Products page in the LiveCode account management site.

It is also possible to deactivate LiveCode with:

```
<livecode> deactivate
```

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

On Windows, the command is:

```
start /wait <livecode> activate -file LICENSE -passphrase SECRET
start /wait <livecode> deactivate
```

On Mac OS X, you need to do:

```
<livecode>/Contents/MacOS/LiveCode activate -file LICENSE -passphrase SECRET
<livecode>/Contents/MacOS/LiveCode deactivate
```

LiveCode Community engine changes

Android 6.0 runtime permissions

Android 6.0 (API 23) Marshmallow introduced a new permissions model that lets apps request permissions from the user at runtime, rather than prior to installation. Apps built with LC 9.0.1 do support this new permissions model, and request permissions automatically when the app actually requires the services or data protected by the services.

For example, if the app calls `mobilePickPhoto "camera"`, a dialog will be shown to the user asking for permission to access the device camera.

If the user does not grant permission, the call will fail. Moreover, the app can use the function `androidRequestPermission(permissionName)` to check if the permission for `permissionName` has been granted.

Notes:

- You have to make sure that you check the required permissions for your app in the standalone settings.
- Apps that run on devices running Android 6+ will work with the new permissions model.
- Apps that run on older devices (less than Android 6) will continue to work with the old permissions model.
- If the user does not grant a permission when the dialog appears for the first time, they can change this preference from the Settings app.

New stereo panning properties added to MacOSX Player object

leftBalance: control the volume of the left stereo channel. rightBalance: control the volume of the right stereo channel. audioPan: pan audio from one stereo channel to another.

Note: these properties currently require media files to have stereo audio tracks. There will be no effect on mono or surround-sound formats.

Support resizing stacks for orientation changes in fullscreen modes

A new mobile command `mobileSetFullScreenRectForOrientations` has been implemented to allow stacks that use the `fullscreenMode` property to be resized when the device orientation changes.

Specific engine bug fixes (9.0.1-rc-3)

21532 Fix dragData clearing between mouseDown and dragStart

Specific engine bug fixes (9.0.1-rc-2)

21396 Fix crash on startup in iOS 12 beta
21417 Don't include any externals in emscripten standalones
21434 Fix visual effects not working when acceleratedRendering is true on Android
21443 Fix groups sometimes not redrawing when scrolled
21496 Ensure emscripten aux stacks are loaded on startup

Specific engine bug fixes (9.0.1-rc-1)

10694 Ensure the standalone builder throws an error if the Android SDK path is not set or is invalid
12431 Ensure .otf font files are recognized on iOS and Android
12675 Ensure sound can be queued with mobilePlaySoundOnChannel on Android
13266 Ensure option menu works on Android
15086 Included mention of usePixelScaling's error throwing behaviour on Android.

15443	Ensure caseless comparison of native strings works on Mac
15614	Fixed incorrect example in the dictionary entry for URLEncode.
16366	Ensure the area outside the stack rect is re-rendered when going to a new card in fullscreen mode showAll
16736	Fix crash when relayering objects with accelerated rendering true
19094	Ensure single-codepoint grapheme clusters are checked for font support
19465	Change the orientation on Android when `mobileSetAllowedOrientations` does not allow the current orientation
19651	Windows: Fix incorrect video output for some formats when playing mirrored
19900	Ensure tsNet external can be unloaded
20016	DataGrids completely unresponsive in HTML 5 standalones
20211	Fix Mac player crash when setting filename multiple times
20550	Update ExtVideoView module to compile against API 19
20551	Android: Update Notification module to compile against API 23
20552	Android: Replace deprecated method of transferring file data to/from other apps
20570	revDB: Fix MySQL errors after calling stored procedures
20631	Fix crash when launching or resuming android app from local notification
20678	Ensure that when the clipboard only has private data it will be updated when the system clipboard changes
20839	Improve appearance of disabled buttons when using Motif look-and-feel
20925	Fix crash when repeatedly dragging
20997	Add tsNet builds for iOS SDK 11.3
21019	Support resizing stacks for orientation changes in fullscreen modes
21049	Add appropriate errors for `the fontLanguage`
21051	Ensure revDataFromQuery on an ODBC database does not return incomplete unicode strings
21062	Fix detection of HTTP digest authentication with tsNet
21141	Fixed errors when setting textFont/textSize on Android native button
21155	Fix JavaScript error when closing stack window in HTML5 standalone
21161	LiveCode trials do not start on Windows
21171	Ensure unicode characters in app label are shown correctly on Android
21184	Ensure play command can play a remote auriofile on iOS
21207	Ensure 'relaunch' handler is found when declared in parentscript
21217	Prevent crash when calling quit from a stack with a non empty imagesource
21228	Ensure android working screenrect is updated when the keyboard is presented
21234	Dictionary: revDataFromQuery / revQueryDatabase updates
21246	Fix crash using windows clipboard
21250	Fix crash which may occur when selected objects are deleted
21253	Build iOS 11.4 binaries for the tsNet external
21271	Fix crash when setting the markerpoints of a graphic in a repeat loop
21278	Ensure playrate is respected when looping and when playing after a pause
21279	Ensure objects are clipped correctly when rendering fullscreen mode leterbox with acceleratedRendering true
21280	Fix crash when setting the styledText of a long field to its styledText
21302	Make message box 'put' statements immune to 'lock messages'

21304 Clear unshared data when compacting stack
21317 Ensure Search For Inclusions works on both device and simulator
21321 Fix crash when getting the hostnametoaddress without param
21326 Ensure binary strings remain so when binary string is appended or prepended
21336 Ensure only the top stack will render on mobile
21352 Android: fix potential crash on relaunch due to uninitialized static variable
21356 Fix black screen on Android when setting acceleratedRendering at startup
21368 Ensure push notifications work on Android when targetSdkVersion=26
21386 Fix widgets with native layers not being clipped by their parent group rect on iOS
21395 Fix MacOSX player control not starting when playRate property is 0
21398 Do not split up .framework folders between MacOS and Resources/_MacOS folder

LiveCode Community IDE changes

Specific IDE bug fixes (9.0.1-rc-1)

18585 Ensure Dictionary does not pass cmd+A
19504 Allow entry of tabs into the text property field of the property inspector
19998 Add default points for polygon graphic
20094 Fix multi-line message box not executing if the first line is a comment
20672 Fix very slow arrow key nudge of multiple objects
21017 Add `resizeControl` to the list of handlers not to trace in the debugger as doing so locks up the IDE
21086 Keep correct selection when formatting whole script
21122 Fix user stacks opening offscreen when last opened on a different monitor
21167 Ensure Replace history is remembered
21169 Ensure splash screen always hides on Windows
21172 Fix revMail on mobile
21174 Ensure "Sample Stacks" window does show up in menu "Windows"
21176 Make sure the Start Center can always show the Upgrade Options
21179 Add 'show documentation' option to contextual menu in extension manager
21202 Fix deselection of next find after replace in Script Editor
21206 Fix execution error opening message box from script editor via Cmd/Ctrl+M
21222 Ensure File -> Close option is disabled when the topstack is stack revMenubar
21232 Fix wandering breakpoints when undoing paste
21323 correction to syntax for DG2's new props

LiveCode Community extension changes

Specific extension bug fixes (9.0.1-rc-1)

19753 Prevent chunk out of range error on Linux when clicking selection
20142 Prevent index errors when expanding TreeView buffer beyond 1000 keys
21203 Only post hiliteChanged when value actually changes
21345 Ensure the "horizontal" property of segmented control widget is saved

LiveCode Community Plus IDE changes

Specific IDE bug fixes (9.0.1-rc-1)

20975 Obscure pasted and dragged passwords in the activation dialog

LiveCode Indy extension changes

Specific extension bug fixes (9.0.1-rc-1)

21125 Fix missing docs references

LiveCode builder changes

LiveCode Builder Virtual Machine

Array and list assign ops

Previously there was a difference between constructing a list or array using `push` or `put` and using list or array assigment expressions `[]` and `{}`, namely values were converted to `optional any` only in the latter case. For consistency, they are now converted in both cases.

LiveCode Builder Language

Nullable aggregate fields

- Aggregate fields of Pointer type can now contain `nothing`, i.e. null pointers

Specific builder bug fixes (9.0.1-rc-1)

20931 Values should bridge to optional any in array and list assign

21064 Ensure multi-module assembly manifest requires encompass support module use clauses

21297 Fix issue with importing foreign value types that require conversion

Dictionary additions

- **androidHasPermission** (*function*) has been added to the dictionary.
- **androidPermissionExists** (*function*) has been added to the dictionary.
- **androidRequestPermission** (*command*) has been added to the dictionary.
- **audioPan** (*property*) has been added to the dictionary.
- **leftBalance** (*property*) has been added to the dictionary.
- **messageDigest** (*function*) has been added to the dictionary.
- **rightBalance** (*property*) has been added to the dictionary.

Previous release notes

- LiveCode 9.0.0 Release Notes
- LiveCode 8.1.9 Release Notes
- LiveCode 8.1.8 Release Notes
- LiveCode 8.1.7 Release Notes
- LiveCode 8.1.6 Release Notes
- LiveCode 8.1.5 Release Notes
- LiveCode 8.1.4 Release Notes
- LiveCode 8.1.3 Release Notes
- LiveCode 8.1.2 Release Notes
- LiveCode 8.1.10 Release Notes
- LiveCode 8.1.1 Release Notes
- LiveCode 8.1.0 Release Notes
- LiveCode 8.0.2 Release Notes
- LiveCode 8.0.1 Release Notes
- LiveCode 8.0.0 Release Notes
- LiveCode 7.1.4 Release Notes
- LiveCode 7.1.3 Release Notes
- LiveCode 7.1.2 Release Notes
- LiveCode 7.1.1 Release Notes
- LiveCode 7.1.0 Release Notes
- LiveCode 7.0.6 Release Notes
- LiveCode 7.0.4 Release Notes
- LiveCode 7.0.3 Release Notes
- LiveCode 7.0.1 Release Notes

- LiveCode 7.0.0 Release Notes
- LiveCode 6.7.9 Release Notes
- LiveCode 6.7.8 Release Notes
- LiveCode 6.7.7 Release Notes
- LiveCode 6.7.6 Release Notes
- LiveCode 6.7.4 Release Notes
- LiveCode 6.7.2 Release Notes
- LiveCode 6.7.11 Release Notes
- LiveCode 6.7.10 Release Notes
- LiveCode 6.7.1 Release Notes
- LiveCode 6.7.0 Release Notes
- LiveCode 6.6.2 Release Notes
- LiveCode 6.6.1 Release Notes
- LiveCode 6.6.0 Release Notes
- LiveCode 6.5.2 Release Notes
- LiveCode 6.5.1 Release Notes
- LiveCode 6.5.0 Release Notes
- LiveCode 6.1.3 Release Notes
- LiveCode 6.1.2 Release Notes
- LiveCode 6.1.1 Release Notes
- LiveCode 6.1.0 Release Notes
- LiveCode 6.0.2 Release Notes
- LiveCode 6.0.1 Release Notes
- LiveCode 6.0.0 Release Notes