LiveCode 8.1.4-rc-3 Release Notes

Overview

LiveCode 8.1 provides important improvements for delivering high-quality cross-platform applications!

- LiveCode Indy and Business editions now come with the tsNet external, which supercharges LiveCode's Internet features and performance. LiveCode 8.1 also introduces mergHealthKit, for accessing activity, sport and health data on iOS devices.
- The standalone builder now has a greatly-improved user experience for including externals, script libraries and LiveCode Builder extensions in your cross-platform application. Usually, it'll now do the right thing automatically, but you can still select the specific inclusions you need.
- The IDE has lots of other upgrades, too: a keyboard-navigable Project Browser that highlights any scripts that failed to compile, an improved dictionary user interface, and access to the message box just by starting to type.
- The player control can be used in Windows application without any need for users to install any additional libraries or dependencies, thanks to a brand new player implementation based on DirectShow. For most apps, it should now be unnecessary to install or use QuickTime at all.
- The LiveCode Builder programming language has had some enhancements as part of the Infinite LiveCode project. Variables now get initialised by default, `unsafe` blocks and handlers can be used to flag sections of code that do dangerous things, and you can even include raw `bytecode` if necessary.

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 or the revVideoGrabber external.

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 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)
- 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:

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

- 10.8.x (Mountain Lion) on Intel
- 10.9.x (Mavericks) on Intel
- 10.10.x (Yosemite) on Intel
- 10.11.x (El Capitan) on Intel
- 10.12.x (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 4.6 on MacOS X 10.7
- Xcode 5.1 on MacOS X 10.8
- 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 8.3 on MacOS 10.12

It is also possible to set other versions of Xcode, to allow testing on a wider range of iOS simulators. For instance, on OS X 10.10 (Yosemite), you can add *Xcode 5.1* in the *Mobile Support* preferences, to let you test your stack on the *iOS Simulator 7.1*.

We currently support deployment for the following versions of iOS:

- 6.1 [simulator]
- 7.1 [simulator]
- 8.2 [simulator]
- 9.2
- 10.2
- 10.3

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 ARMv6, ARMv7 or ARMv8 processors. It will run on the following versions of Android:

- 2.3.3-2.3.7 (Gingerbread)
- 4.0 (Ice Cream Sandwich)
- 4.1-4.3 (Jelly Bean)
- 4.4 (KitKat)
- 5.0-5.1 (Lollipop)
- 6.0 (Marshmallow)

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
```

Engine changes

Fix deletion of the target in safe cases (8.1.4-rc-2)

You can now safely 'delete the target' as long as there are no handlers on the stack owned by the target.

After deleting 'the target', 'the target' will become empty which will result in an execution error when an attempt is made to dereference it.

Ensure browser widgets are in the correct location (8.1.4-rc-2)

Browser widgets in nested groups now remain in the correct location rather than shifting down vertically out of sync with everything else.

Script-only deploy (8.1.4-rc-1)

It is now possible to use script-only stacks in the mainstack and auxiliary stack parameters to the deploy command.

Specific engine bug fixes (8.1.4-rc-3)

19730 Fix resolution of relative paths of images in Mac standalones

Specific engine bug fixes (8.1.4-rc-2)

19026 Fix DNS resolver issue causing connection hang when using tsNet on Linux
19525 Fix SFTP connection hang when remote directory doesn't exist
19573 Provide iOS 10.3 builds for tsNet
19577 Fix crashes related to card deletion
19578 Fix crash after leaving edit group mode
19584 Allow deletion of a message's target object in a frontscript
19587 Fix deletion of the target in safe cases
19592 Prevent crash when deleting combo box
19635 Ensure browser widgets are in the correct location

Specific engine bug fixes (8.1.4-rc-1)

10947 Fix hypercard-compatibility dynamic path behavior
11170 Disable keyboard suggestion when entering password in Android native input field.
11727 Fix arrow key not work in Android field
12187 Make sure keyboardActivated/keyboardDeactivated messages are sent when the status bar is hidden on Android
14238 Ensure background pattern stays aligned in long fields
17577 Make sure we can set the hilitedItemName property of the navBar widget
18058 Fix keyboard not show in landscape orientation
18273 Prevent crash when rendering card with invalid objptr
Make sure mobileControlGet does not return rounded values of

18358 startTime/endTime/currentTime
18454 Allow socket to send broadcast packet on Android.
18539 Don't change the defaultFolder on startup
18619 Delete slash at the end of specialFolderPath("resources") to be consisten with other result of special folders.
18833 Don't change name of tsNet stack during standalone build
18912 Ensure objects can't be deleted if their behaviors are handling a message or they are the target
18946 Fix browserNavigateComplete not firing when document has frames
19060 Ensure error when binding widget is caught correctly
19116 Ensure tsNetGetStatus reports transfer status as "uploading" appropriately during POST requests
19138 Update OpenSSL to version 1.1.0d
19154 Fix widget browser stuck on handling javascript
19192 Add ios and android to "allowDatagramBroadcasts" dictionary entry
19200 Make sure printSettings are set correctly
19212 Prevent a crash when calling mobileComposeMail with just one param (tSubject)
19215 Make sure botton icons are present in standalones when building for multiple platforms
19229 Fix crash when connecting an IR Receiver
19246 Remove objects from message path ASAP when deleted
19279 Prevent mobilePickMedia crash without 'Write External Storage' permission
19287 Make sure the clickLoc is updated on mouseDown/touchEnd on mobile
19293 Server returns 'ELF' over HTTP
19298 Make sure "Search for Inclusions" detects correctly widget inclusions on iOS simulator
19307 Prevent crash when saving standalone while player is playing
19313 Fix crash when saving field with fdata
19320 Fix SVG parsing of 'a' instruction
19328 Standalone startup stack needs to be mainstack before resolving parent scripts
19352 Fix crash when getting the urlResponse
19390 Reposition native layer controls correctly when resizing stacks
19404 Fix crash on iOS when calling `play empty` followed by `play path/to/audio/file`
19416 Ensure all object messages are cleared when obj or ancestor is deleted
19417 Use correct comment syntax in docs on documentation
19424 Make sure getting `the securityPermissions` returns the expected result
19457 Prevent crash when deleting selected objects with the backspace key
19469 Make sure the long time format is correct if `twelveHourTime` is false
19490 Make sure launch url sets the result to empty on success
19500 Fix crash when invalid url is used with iOS native browser
19515 Ensure the formattedRect of line N always returns the correct result

IDE changes

Specific IDE bug fixes (8.1.4-rc-2)

19564 Prevent error when deleting script editor tab

Specific IDE bug fixes (8.1.4-rc-1)

17448 Make sure messages are sent when going to stacks/cards from the Project Browser
18035 Make sure the gradient popup stack is displayed as expected
18485 Ensure relayering menu items don't relayer objects out of owner groups
18549 Make sure `lock cursor` works in the IDE
18991 Disable custom property editor when no node selected
19152 Show warning if the new stack name begins with "rev"
19153 Ensure objects can not be dragged to invisible open stacks from the tools palette
19160 Make sure the S/B respects the "iPad initial orientations" settings
19177 Update guide images for LiveCode 8
19178 Add test to ensure default folder doesn't change when loading IDE
19179 Add tests for standalone builder inclusions
19181 Ensure tutorial has location set when being skipped
19188 Make outputting debug vars from message box work in all contexts
19195 Allow vertical scrolling in "Value" field of Variable Visualizer window
19196 Ensure extension is installed before deleting files
19439 Ensure the "Effects" settings stack always appears onscreen
19451 Don't try to delete breakpoint while it is a target in the call stack

LiveCode Builder changes

LiveCode Builder Standard Library

Foreign function interface

- When passing a Number to one of the foreign integer types (`LCInt`, `LCUInt`, `IntSize`, `UIntSize`), an error will be thrown if the value is outside the range of the requested type.
- The `IntSize` and `UIntSize` types can hold the full 64-bit integer range, however the maximum magnitude which is supported for converting to and from Number is 2^53. An error will be thrown for any conversions outside this range.

Specific LCB bug fixes (8.1.4-rc-1)

19067 Ensure an error is thrown if there is no script access
19214 Increase usable range of IntSize and UIntSize types
19244 Nil pointers should bridge to nothing

LiveCode extension changes

Specific extension bug fixes (8.1.4-rc-1)

19261 Clear selection when deleting selected node
19350 Remember iconPresetName in SVG Icon when reopening

Previous release notes

- LiveCode 8.1.3 Release Notes
- LiveCode 8.1.2 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