LiveCode 8.1.3 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 8.2 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

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

Ensure CMYK JPEGs display correctly on Mac (8.1.3-rc-1)

This fixes the incorrect rendering of CMYK JPEGs containing an ICC profile on Mac.

Add builtin implementation of field 'Select All' (8.1.3-rc-1)

The field control will now understand Cmd/Ctrl+A as the 'Select All' action - causing the whole text of the field to be selected.

Improve efficiency of compiled regex cache (8.1.3-rc-1)

The efficiency of lookups of previously compiled regexs has been improved. To take full advantage

of the regex cache make sure that you either use a string constant for the regex pattern, or a variable which is not mutated between uses. e.g.

```
get matchText(tTarget, "someregexpattern") -- efficient
get matchText(tTarget, tUnchangedPatternVar) -- efficient
get matchText(tTarget, tPatternPrefix & tPatternSuffix) -- inefficient
```

In general you will only gain advantage from the regex cache if you repeated use the same regex pattern in the way described above repeatedly in a tight loop.

Ensure matchText and replaceText don't affect target string (8.1.3-rc-1)

Previously calling matchText or replaceText on a string would cause subsequent uses of that string to use slower codepaths causing unexpected performance degredation.

Specific engine bug fixes (8.1.3-rc-2)

19105	Resolve crash when deleting a stack in the message path in a front script
19120	Ensure substacks with cantDelete don't cause execution errors when deleting the mainstack
19121	Pass correct folder to standaloneSaved message
19158	Prevent crash when using undo to bring back removed controls
19185	Ensure the dragData["private"] isn't cleared during a drag

Specific engine bug fixes (8.1.3-rc-1)

11313	Support mobilePickPhoto() maximum width and height settings on Android
13055	Improve formatting of try syntax description
13151	Correct example of setting "listIndent" for whole field
13370	Correct terminology in "convert" command documentation
14080	Fix find command not finding in specified field when not on current card
17918	Make sure setting clipboarddata["text"] and clipboarddata["html"] works as expected
17973	Make sure `the machine` can distinguish between iOS device and simulator
18264	Don't fail standalone build completely if unlicensed platforms are selected
18277	Calculate the height of the mac desktop space correctly
18295	Fix empty specialFolderPath("resources") on Windows
18353	Remove duplicated urlResponse documentation
18392	Ensure "ceiling" is listed as a reserved word
18443	Ensure 'private' clipboard is reset when another application sets the clipboard
18459	Fix incorrect behavior of files() and folders() function on Android.
18514	Make sure setting clipboarddata["html"] works as expected on Windows
18600	Fix crash when quitting from script editor
18610	Add code examples to tsNet dictionary entries

18632	Mark the copyResource function as deprecated
18652	Fix occasional crash when getting the clipboarddata["text"] on Windows.
18666	Fix crash when find command matches text in sharedText field on non-current card
18686	Fix a crash related to the message watcher
18724	Fix incorrect cross-references in lockLocation dictionary entry
18738	Fix data loss when cr inserted into a styledText run
18755	Fix loss of BMP as supported clipboard image format
18762	Fix a rare crash on saving after cloning a field
18778	Send standalone saving messages at correct time
18852	Fix exception thrown in IDE when saving standalone with more than one stack
18870	Fix crash caused by calling tsNetGetSync() repeatedly against the same URL
18871	Add support for upper and lower case field shortcuts on Mac
18873	Fix documentation of valid values for lineSize property
18890	Fix crash when playing non-imported audioclip
18893	Fix formatting in description of stack mode property
18894	Don't strip context tags from HTML clipboard formats
18923	Ensure CMYK JPEGs display correctly on Mac
18925	Prevent crashes on memory exhaustion
18938	Fix text encoding issues when pasting HTML into LiveCode
18948	Make 'obj of me' consistent across all control types
18959	Fix crash after getting the points of a regular polygon graphic
18969	Add builtin implementation of field 'Select All'
18977	Ensure setting the card triggers openCard / closeCard in no UI mode
18982	Improve shift+click behavior of text selection in XPDF
18988	Deal with stack id cache correctly on stack delete
19003	Stackfiles under the mainstack path preserve folder structure in standalone
19004	Improve efficiency of compiled regex cache
19005	Ensure matchText and replaceText don't affect target string
19008	Place modals at their own location when neither the defaultStack or topStack can provide useful relative locations
19009	mergExt external builds for iOS 10.2
19010	Provide iOS 10.2 builds for tsNet
19017	Fix crash when deleting object which is being moved
19024	Prevent crash when calling `mobileStoreRequestProductDetails`
19032	Prevent crash when creating and deleting two data grids
19036	Fix incorrect local file path when setting url of browser widget on Android.
19044	Prevent crash when deleting stack which has an open substack
19045	Fix BiDi algorithm for surrogate pairs
19049	Prevent crash when deleting mousestack stack via button click
19050	CGI does not work with lighttpd 1.4.44
19051	Make sure the player endtime property cannot be set to a value greater than the player duration
19076	Prevent redraw recursion when going to stack twice in no ui mode
19084	Ensure text is copied as unstyled on mac
19088	Do not show "Success" dialog if an error occurred in S/B

19093   Prevent recursion when fetching mouseColor in mouseMove handler

19097   Ensure all menus are updated after mouseDown to menu group

19108   Correct references for scriptOnly property docs

8274   Improve modal placement documentation

IDE changes

Allow substack to become a mainstack via property inspector (8.1.3-rc-1)

The property inspector Basic pane for substacks now has a button beneath the mainstack labelled "Make mainstack". Note that once pressed this button will disappear, as the stack will no longer a substack.

Reinstate store tab of extension manager (8.1.3-rc-1)

The store tab of the extension manager has been reinstated and the revBrowser implementation has been replaced with a browser widget.

Specific IDE bug fixes (8.1.3-rc-1)

15917   Font Size for Project Browser can now be set from LiveCode Preferences -> Project Browser

17889   Repaired confusing layout of fill gradient control in Property Inspector

18177   Reinstate text properties and graphic effects to datagrid inspector

18302   Retain custom prop changes when clicking on tree view in editor

18393   [Project Browser] Change "Sort controls by number" to "Sort controls by layer" to avoid confusion + make sure they are sorted numerically

18491   Allow substack to become a mainstack via property inspector

18586   Make sure the Project Browser stack/card/group view can always expand

18701   Prevent over-enthusiastic save prompts

18726   Make sure the Bug Report checkmark appears at the correct place in Standalone Settings

18804   Update locked inspectors when mainstack names change

18897   "Show Sections" for Project Browser can now be set from LiveCode Preferences -> Project Browser

18930   Reinstate store tab of extension manager

18981   Added tooltip to iOS icon and splash screen selection

18987   Reinstate fixedLineHeight for tableField

19015   Reset the templateStack after datagrid creation

LiveCode Builder changes

Specific LCB bug fixes (8.1.3-rc-1)

18856   Use cached numeric values when converting numbers to/from strings

LiveCode extension changes

Tree View widget

Limit display calculations

Previously when an array was expanded in the Tree View widget, all of the display calculations for were done before the next redraw. Now the keys are sorted (as before) but the display calculations are made for a maximum of 1000 rows. When more rows are needed due to scrolling, another 1000 are calculated at that point. This provides a near-continuous scrolling experience for arrays with large numbers of keys, and ties the expense of expanding an array to that of sorting its keys.

Specific extension bug fixes (8.1.3-rc-1)

18500   Ensure color properties are documented correctly
18693   Prevent long delays when expanding arrays with many keys

Previous 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