

LiveCode 8.1.2-rc-3 Release Notes

- Overview
- Known issues
- Platform support
 - Windows
 - Linux
 - Mac
 - iOS
 - Android
 - HTML5
- Setup
 - Installation
 - Uninstallation
 - Reporting installer issues
 - Activating LiveCode Indy or Business edition
 - Command-line installation
 - Command-line uninstallation
 - Command-line activation for LiveCode Indy or Business edition
- Engine changes
 - Improved ul_TraceLocals output (8.1.2-rc-1)
 - Fix problems with printing PDFs to some printers (8.1.2-rc-1)
 - Specific engine bug fixes (8.1.2-rc-3)
 - Specific engine bug fixes (8.1.2-rc-2)
 - Specific engine bug fixes (8.1.2-rc-1)
- IDE changes
 - Specific IDE bug fixes (8.1.2-rc-2)
 - Specific IDE bug fixes (8.1.2-rc-1)
- LiveCode extension changes
 - Specific extension bug fixes (8.1.2-rc-1)
- Previous 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 8.1 on MacOS X 10.11
- Xcode 8.1 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
- 10.1

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	<code><x86 program files folder>/RunRev/LiveCode <version></code>
Linux	<code>/opt/livecode/livecode-<version></code>

The installations when installing for "This User" are:

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

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
	Install the IDE for "All Users". If not specified, LiveCode will be installed

Option	Description
-allusers	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

Improved `ul_TraceLocals` output (8.1.2-rc-1)

The `ul_TraceLocals` function in `revLibURL` has been improved so that it prints out keys and values of the script local arrays in the library. This is useful for troubleshooting `libURL` issues.

Fix problems with printing PDFs to some printers (8.1.2-rc-1)

It was possible for LiveCode to generate PDFs which were incompatible with some printers. This has been fixed by upgrading the PDF generation library which LiveCode uses (`cairo`).

Specific engine bug fixes (8.1.2-rc-3)

- 18900** Fix a crash when closing a stack with substacks still open
- 18911** Fix graphical artefacts when reshaping polygon graphics while selected
- 18958** Make sure our prebuilt libs do not use reserved (by Apple private APIs) function names

18962 Fix crash when using field

Specific engine bug fixes (8.1.2-rc-2)

- 18484 Prevent mobilePlaySoundOnChannel crash on Android
- 18809 Prevent lock up of PI when not selecting choice from font menu
- 18810 Fix a crash when cutting controls
- 18812 Fix crash when opening cards referencing images on non-open cards
- 18823 Fix a crash when popping up transient windows
- 18824 Fix a crash when drag-selecting controls
- 18863 Fix encrypt/decrypt operations broken by the OpenSSL 1.1.0 update

Specific engine bug fixes (8.1.2-rc-1)

- 15384 Fix incorrect handling of a 204 response from a server.
- 16965 Fix incorrect placement of browser widget after stack rect change when fullscreenmode used.
- 17008 Fix selection handles remaining after selected object deleted
- 17247 Remove selection artefacts when handles are drawn outside of parent group rect
- 17541 Fix problems with printing PDFs to some printers
- 17622 Fix extra data added on Windows when pasting html data from the clipboard
- 17779 Fix scrolling group drawing outside its bounds when acceleratedRendering used.
- 18275 Enable sqlite FTS5 feature
- 18287 Update sqlite version to 3.15.0
- 18293 Crash when deleting a stack that is used as a popup menu
- 18343 Fix incorrect result from itemOffset when first character of stringToSearch is the delimiter
- 18349 Fix variable contents modified when used to set stack name
- 18379 Don't include incorrectly copied resource forks in standalones
- 18406 Fix delay in triggering handlers when called by JavaScript in browser widget
- 18440 Respect SB Copy Files pane relative / absolute path distinction on mobile
- 18441 Make sure the purchaseStateUpdate callback is sent with status=complete when necessary
- 18444 Make sure put cookie with empty value works as expected
- 18472 'load url' is not properly cleaned up on socketError
- 18473 Prevent hang when adjusting field pixmap offset
- 18488 Error returned by hostnametoaddress was not being reported in libURL.
- 18496 Fix memory leak when using filter on unicode strings
- 18498 Ensure bundled Android externals are available on Windows and Linux
- 18499 Fix libUrlSetStatusCallback on mobile platforms when tsNet is in use
- 18521 Resolve folder path before processing files(folder) and folders(folder)
- 18536 Added support for Xcode 8.1 / iOS 10.1
- 18566 libURL inserts "://" between host and port when creating CONNECT request
- 18578 Ensure color name rgb value mapping is in alphabetical order

- 18604 Fix crash when converting objective-c objects to LiveCode values
- 18614 Fix Linux player crash when accessing properties
- 18623 Remove tsneterr: from the start of output of tsNetVersion()
- 18625 Fix browser javascripthandlers not working on iOS after loading a new page
- 18626 Make sure the Standalone Application Settings on Windows are respected
- 18642 Fixed crash on iOS 10 when trying to read local notifications
- 18653 Encode bundle display name as utf-8 in ios app plist
- 18683 Fix crash on iOS 10 when the app needs access to the device's microphone
- 18690 Provide mergExt Builds for building against iOS 10.1 SDK
- 18691 Provide tsNet Builds for building against iOS 10.1 SDK
- 18703 Fix handling of new tsNet network requests when an existing network request is still waiting to connect
- 18709 Cannot deploy an app to iOS 10.1 simulator
- 18783 Resolve delay in opening Windows standalones that include tsNet 1.2.4 and the Internet inclusion
- 6506 Fix regression to watching global variables

IDE changes

Specific IDE bug fixes (8.1.2-rc-2)

- 18857 Import as control > Text file doesn't set text of control

Specific IDE bug fixes (8.1.2-rc-1)

- 17447 Reinstate resize checkbox in property inspector Position pane
- 17536 Fix or mitigate effect of nudging many controls at once
- 17618 Reinstate 'fit content' button in property inspector Position pane
- 18029 Show 20 fonts at a time in property inspector font menu
- 18290 Use datagrid template safely while building custom headers
- 18292 Fix hidden palettes not reappearing
- 18300 property inspector custom property list is not sorted
- 18452 Saving a substack from the Project Browser no longer asks for a path to save
- 18455 Show the correct version of LiveCode in Start Center title
- 18460 Mark stack as edited when property changed from the PI
- 18483 Prevent iOS display name standalone setting becoming utf-8 encoded data
- 18557 Ensure dragging object from tools palette is smooth
- 18721 Make sure unchecking "Notify me of development releases" is respected
- 18791 Fix PI list editors not updating when value changed

LiveCode extension changes

Specific extension bug fixes (8.1.2-rc-1)

- 18319 Prevent segmented control fill from bleeding outside border.
- 18391 Correctly order default marker styles

Previous 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](#)