

LiveCode 8.1.0-dp-1 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](#)
 - [Automatic LCB extension inclusion in standalones \(8.1.0-dp-1\)](#)
 - [Mobile search for inclusions \(8.1.0-dp-1\)](#)
 - [Mobile script library inclusions \(8.1.0-dp-1\)](#)
 - [Windows DirectShow Player Control \(8.1.0-dp-1\)](#)
 - [Add support for custom entitlements for iOS \(8.1.0-dp-1\)](#)
 - [Specific engine bug fixes](#)
- [IDE changes](#)
 - [Standalone Inclusions Interface \(8.1.0-dp-1\)](#)
 - [Specific IDE bug fixes](#)
- [LiveCode Builder changes](#)
 - [LiveCode Builder Language](#)
 - [Specific LCB bug fixes](#)
- [LiveCode extension changes](#)
 - [SVG Icon widget](#)
- [Previous release notes](#)

Overview

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

- The standalone builder now has a greatly-improved user experience for including externals, script libraries and LiveCode Builder extensions in your cross-platform application.
- The player control can be used in Windows application without any need for users to install any additional libraries or dependencies.

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 Linux installations which meet the following requirements:

- Supported CPU 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

- 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)
- Requirements for 32-bit Intel/AMD:
 - glibc 2.11 or later
- Requirements for 64-bit Intel/AMD:
 - glibc 2.13 or later
- Requirements for ARMv6:
 - glibc 2.7 or later

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

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 7.3 on MacOS X 10.11

It is also possible to set other versions of Xcode, to allow testing on a wider range of iOS simulators. For instance, on 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 the following iOS Simulators:

- 6.1
- 7.1
- 8.2
- 9.2
- 9.3

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.

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 yet 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	<code><documents and settings folder>/<user>/Local Settings/</code>
Windows Vista/7	<code><users folder>/<user>/AppData/Local/RunRev/Logs</code>
Linux	<code><home>/.runrev/logs</code>

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
<code>-allusers</code>	Install the IDE for "All Users". If not specified, LiveCode will be installed for the current user only.
<code>-desktopshortcut</code>	Place a shortcut on the Desktop (Windows-only)
<code>-startmenu</code>	Place shortcuts in the Start Menu (Windows-only)
<code>-location LOCATION</code>	The folder to install into. If not specified, the <code>LOCATION</code> defaults to those described in the "Installation" section above.
<code>-log LOGFILE</code>	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

Automatic LCB extension inclusion in standalones (8.1.0-dp-1)

When a standalone is built, the modules required for the widgets that are on the stack (or any of its substacks) are now included in the application automatically, regardless of whether search for inclusions is selected in the standalone settings.

If search for inclusions is selected, the scripts of the application will be searched for uses of the public handlers of any available LCB libraries, and any uses of the 'kind' of available widgets to determine whether the relevant modules are included. For example, if the script contains

```
create widget as "com.livecode.widget.svgpath"
```

then the SVG Path widget module and all its dependencies will be included.

Mobile search for inclusions (8.1.0-dp-1)

The standalone builder 'search for inclusions' mechanism will now work for mobile deployment, both to device and simulator.

Mobile script library inclusions (8.1.0-dp-1)

Script libraries can now be included in mobile applications in the same way as for desktop applications, via the Inclusions pane of the standalone builder.

Windows DirectShow Player Control (8.1.0-dp-1)

Due to the recent decision by Apple to end support for QuickTime on Windows, the player implementation on that platform has been replaced with one based on DirectShow. This is a multimedia API that is available by default on all versions of Windows supported by LiveCode.

The new implementation should function as a drop-in replacement for the old one, though some properties are not yet implemented.

Property Changes

On Windows, the behaviour of some properties of the player control have changed.

- The **loadedTime** property previously did not work on Windows, but now does.
- The **alwaysBuffer**, **callbacks**, **enabledTracks**, **mediaTypes**, **mirrored**, **trackCount** and **tracks** properties do not currently work, but will be re-enabled in a subsequent release.

On all platforms, the following player control properties, which are specific to QuickTime and QTVR, have been deprecated: **constraints**, **currentNode**, **movieControllerId**, **nodes**, **pan**, **tilt**, and **zoom**.

Supported File Formats

Media format support in the new Windows player control depends on which codecs are installed.

A list of the [file formats and compression types available as standard.aspx](#) on Windows is available in the MSDN documentation

Add support for custom entitlements for iOS (8.1.0-dp-1)

Custom entitlements can now be added to an iOS app by including one or more `.xcent` files in the copy files section of the standalone builder containing an XML snippet of key/value pairs. For example, if you wanted to add the entitlement for HomeKit to your app you might create a file named `HomeKit.xcent` with the following content:

```
<key>com.apple.developer.homekit</key>
<true/>
```

Specific engine bug fixes

- **Bugs fixed in 8.1.0-dp-1: 17384, 17515, 17573, 17578, 17624, 17652, 17666.**

IDE changes

Standalone Inclusions Interface (8.1.0-dp-1)

The standalone settings user interface has been reworked to unify the notion of app inclusion. There is now an 'Inclusions' pane which allows the user to select from a complete list of available inclusions. The list contains information about which platforms are supported.

The inclusions pane significantly improves the cross-platform development experience provided by LiveCode (since the iOS and Android panes no longer have separate check boxes for the various built-in externals such as revxml), as well as paving the way for much better extensibility in the future.

Specific IDE bug fixes

- **Bugs fixed in 8.1.0-dp-1: 17189, 17441, 17566, 17638, 17669.**

LiveCode Builder changes

LiveCode Builder Language

Variables

- Typed variables are now initialised by default to a suitable empty value. For example:

```
variable tList as List
push "element" onto tList
```

Untyped and `optional` variables are initialised to `nothing`.

Specific LCB bug fixes

- **Bugs fixed in 8.1.0-dp-1: 14659.**

LiveCode extension changes

SVG Icon widget

Properties

- New **scaledWidth** and **scaledHeight** properties have been added. These are read-only values that expose the effective size of the rendered SVG path independent of the widget size. When `maintainAspectRatio` is false, then these values are equal to the width and height of the widget.

Previous 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](#)