LiveCode 6.7.7-rc-4 Release Notes

Table of contents

Overview
Known issues
Platform support
  Windows
  Linux
  Mac
Setup
  Installation
  Uninstallation
Reporting installer issues
Activation
Multi-user and network install support (4.5.3)
Command-line installation
Command-line activation
Proposed changes
Engine changes
  Multimedia on MacOS with AVFoundation
  Objects are only deleted on idle
  Ability to set the dontUseQT property for a player object (windows only)
  Default buttons lose their glow when another button is clicked on Yosemite.
  Font mapping feature
  Font mapping file
  Relocation of resources for Mac standalone applications
  Add revBrowser error callback messages.
    Callback messages:
    New iOS status bar style
    [[ In App Purchase ]] Calling mobileStoreRestorePurchases when there are no previous purchases to restore
  Voice Over support
  Cocoa Support
Location Services Disabled with LC 6.6.4 (rc1)
iOS 8 Support
Copy files do not work with the iOS 8 simulator
Non-executable file redirection on Mac
Determining the edition of the running engine
Sound recording returns error message
Threaded Rendering
Effective points of graphics
Player messages aren't sent correctly.
Showing a modal dialog confuses mouse state.
Nine-way stretch for images
Updated text rendering for iOS and OS X
Export snapshot with metadata
New variant of open and secure socket
Multiple density image support for patterns. 
QT-related features don't work. 
Objects which are adjacent don't necessarily appear so at non integral scale factors. 
Queuing too many pending messages causes slowdown and random crashes. 
Inconsistencies in behavior when doing 'delete the selectedChunk'. 
In-App Purchasing 
Clipboard data 'styledText' array accessor. 
Improved revBrowser external 
  JavaScript integration 
    Example: 
OS 10.5 (Leopard) Support 
Setting the label of an option or combo-box does not update the menuHistory. 
Specific bug fixes (6.7.7-rc-4) 
Specific bug fixes (6.7.7-rc-3) 
Specific bug fixes (6.7.7-rc-2) 
Specific bug fixes (6.7.7-rc-1) 
Specific bug fixes (6.7.6-rc-3) 
Specific bug fixes (6.7.6-rc-2) 
Specific bug fixes (6.7.6-rc-1) 
Specific bug fixes (6.7.5-rc-3) 
Specific bug fixes (6.7.5-rc-1) 
Specific bug fixes (6.7.4) 
Specific bug fixes (6.7.4-rc-3) 
Specific bug fixes (6.7.4-rc-1) 
Specific bug fixes (6.7.2) 
Specific bug fixes (6.7.2-rc-2) 
Specific bug fixes (6.7.2-rc-1) 
Specific bug fixes (6.7.1) 
Specific bug fixes (6.7.1-rc-4) 
Specific bug fixes (6.7.1-rc-3) 
Specific bug fixes (6.7.1-rc-2) 
Specific bug fixes (6.7.1-rc-1) 
Specific bug fixes (6.7.0) 
Specific bug fixes (6.7.0-rc-3) 
Specific bug fixes (6.7.0-rc-2) 
Specific bug fixes (6.7.0-rc-1) 
Specific bug fixes (6.7.0-dp-9) 
Specific bug fixes (6.7.0-dp-8) 
Specific bug fixes (6.7.0-dp-7) 
Specific bug fixes (6.7.0-dp-6) 
Specific bug fixes (6.7.0-dp-5) 
Specific bug fixes (6.7.0-dp-4) 
Specific bug fixes (6.7.0-dp-3) 
Specific bug fixes (6.7.0-dp-1) 
IDE changes 
  HTTPS through proxy 
  Proxy automatic configuration support 
Specific bug fixes (6.7.7-rc-3) 
Specific bug fixes (6.7.7-rc-1) 
Specific bug fixes (6.7.6-rc-1) 
Specific bug fixes (6.7.5-rc-3)
Specific bug fixes (6.7.5-rc-1)
Specific bug fixes (6.7.4-rc-1)
Specific bug fixes (6.7.2-rc-1)
Specific bug fixes (6.7.1)
Specific bug fixes (6.7.1-rc-4)
Specific bug fixes (6.7.1-rc-3)
Specific bug fixes (6.7.1-rc-1)

LiveCode Builder changes

Previous Release Notes
Overview
This document describes all the changes that have been made for LiveCode 6.7.7-rc-4, including bug fixes and new syntax.

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.

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
The engine supports the following Windows OSes:
- 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)

*Note:* On 64-bit platforms the engine still runs as a 32-bit application through the WoW layer.

Linux
The linux engine requires the following:
- 32-bit installation, or a 64-bit linux distribution that has a 32-bit compatibility layer
- 2.4.x or later kernel
- X11R5 capable Xserver running locally on a 24-bit display
- glibc 2.3.2 or later
- gtk/gdk/glib (optional – required for native theme support)
- pango/xft (optional – required for pdf printing, anti-aliased text and unicode font support)
- lcms (optional – required for color profile support in JPEGs and PNGs)
- gksu (optional – required for elevate process support)

*Note:* The optional requirements (except for gksu and lcms) are also required by Firefox and Chrome, so if your linux distribution runs one of those, it will run the engine.

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

*Note:* LiveCode and standalones it builds may work on remote Xservers and in other bit-depths, however this mode of operation is not currently 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

*Note:* The engine runs as a 32-bit application regardless of the capabilities of the underlying processor.
Setup

Installation

Each distinct version has its own complete folder – multiple versions will no longer install side-by-side: on Windows (and Linux), each distinct version will gain its own start menu (application menu) entry; on Mac, each distinct version will have its own app bundle.

The default location for the install on the different platforms when installing for 'all users' are:

- Windows: <x86 program files folder>/RunRev/ LiveCode 6.7.7-rc-4
- Linux: /opt/runrev/livecode-6.7.7-rc-4
- Mac: /Applications/ LiveCode 6.7.7-rc-4.app

The default location for the install on the different platforms when installing for 'this user' are:

- Windows: <user roaming app data folder>/RunRev/Components/LiveCode 6.7.7-rc-4
- Linux: ~/.runrev/components/livecode-6.7.7-rc-4
- Mac: ~/Applications/ LiveCode 6.7.7-rc-4.app

Note: If your linux distribution does not have the necessary support for authentication (gksu) then the installer will run without admin privileges so you will have to manually run it from an admin account to install into a privileged location.

Uninstallation

On Windows, the installer hooks into the standard Windows uninstall mechanism. This is accessible from the appropriate pane in the control panel.

On Mac, simply drag the app bundle to the Trash.

On Linux, the situation is currently less than ideal:

open a terminal

cd to the folder containing your rev install. e.g.

cd /opt/runrev/livecode-6.7.7-rc-4

execute the .setup.x86 file. i.e.

./.setup.x86

follow the on-screen instructions.

Reporting installer issues

If you find that the installer fails to work for you then please file a bug report in the RQCC or email support@runrev.com so we can look into the problem.

In the case of failed install it is vitally important that you include the following information:

- Your platform and operating system version
- The location of your home/user folder
- The type of user account you are using (guest, restricted, admin etc.)
- The installer log file located as follows:
  - Windows 2000/XP: <documents and settings folder>/<user>/Local Settings/
Activation

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.

Multi-user and network install support (4.5.3)

In order to better support institutions needing to both deploy the IDE to many machines and to license them for all users on a given machine, a number of facilities have been added which are accessible by using the command-line.

Note: These features are intended for use by IT administrators for the purposes of deploying LiveCode in multi-user situations. They are not supported for general use.

Command-line installation

It is possible to invoke the installer from the command-line on both Mac and Windows. When invoked in this fashion, no GUI will be displayed, configuration being supplied by arguments passed to the installer.

On both platforms, the command is of the following form:

```bash
<exe> install noui options
```

Here options is optional and consists of one or more of the following:

- **-allusers**: Install the IDE for all users. If not specified, the install will be done for the current user only.
- **desktopshortcut**: Place a shortcut on the Desktop (Windows-only)
- **-startmenu**: Place shortcuts in the Start Menu (Windows-only)
- **location**: The location to install into. If not specified, the location defaults to those described in the Layout section above.
- **log logfile**: A file to place a log of all actions in. If not specified, no log is generated.

Note that the command-line variant of the installer does not do any authentication. Thus, if you wish to install to an admin-only location you will need to be running as administrator before executing the command.

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

In what follows `<installerexe>` should be replaced with the path of the installer executable or app (inside the DMG) that has been downloaded.

On Windows, you need to do:

```
start /wait <installerexe> install noui options
```

On Mac, you need to do:

```
"<installerexe>/Contents/MacOS/installer" install noui options
```
On both platforms, the result of the installation will be written to the console.

**Command-line activation**

In a similar vein to installation, it is possible to activate an installation of LiveCode for all-users of that machine by using the command-line. When invoked in this fashion, no GUI will be displayed, activation being controlled by any arguments passed.

On both platforms, the command is of the form:

```text
<exe> activate -file license -passphrase phrase
```

This command will load the manual activation file from `license`, decrypt it using the given `passphrase` and then install a license file for all users of the computer. Manual activation files can be downloaded from the 'My Products' section of the RunRev customer accounts area.

This action can be undone using the following command:

```text
<exe> deactivate
```

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

In what follows `livecodeexe` should be replaced with the path to the installed LiveCode executable or app that has been previously installed.

On Windows, you need to do:

```text
start /wait <livecodeexe> activate -file license -passphrase phrase
start /wait <livecodeexe> deactivate
```

On Mac, you need to do:

```text
"<livecodeexe>/Contents/MacOS/LiveCode" activate -file license -passphrase phrase
"<livecodeexe>/Contents/MacOS/LiveCode" deactivate
```

On both platforms, the result of the activation will be written to the console.

**Proposed changes**

The following changes are likely to occur in the next or subsequent non-maintenance release:

- The engine (both IDE and standalone) will require gtk, gdk, glib, pango and xft on Linux
Engine changes

Multimedia on MacOS with AVFoundation (6.7.7-rc-1)

What has changed?

The player object until now used QuickTime/QTKit APIs for audio and video playback. Since both QuickTime and QTKit have been deprecated by Apple, we have updated the player to use the new AVFoundation API. AVFoundation does not provide a controller for multimedia playback until OSX 10.9 and their new control bar is also missing some of the features provided by the QTKit controller, which required us to implement our own controller to ensure backward compatibility.

We have added three new properties to the player object enabling you to customise the appearance of the controller:

- The `hilitecolor` of a player is the color of the played area, the colour of the volume area, as well as the background color of a controller button when it is pressed.

- The `forecolor` of a player is the color of the selected area. The selected area is the area between the selection handles.

- The `backcolor` of a player is the color of the controller icons (volume icon, play/pause icon, scrub back/scrub forward icon).

We have also added support for getting information about the download progress of a remote multimedia file:

- The `loadedtime` of a player is the time up to which the movie can be played. The download progress is also displayed on the controller well.

You can also query the `status` property of the player. This property can take either of the values:

- `loading` (for remote multimedia files)
- `playing`
- `paused`

A new message is added to the player:

- The `playRateChanged` message is sent to the player when the rate is changed by the rate scrollbar controller. To enable the rate scrollbar controller, hold shift + click on scrubForward/scrubBack buttons of the player controller.

Note AVFoundation player is supported in OSX 10.8 and above. On systems running OSX 10.6 and 10.7, LiveCode continues to provide player functionality using the QTKit API.

Objects are only deleted on idle (6.7.7-rc-1)

The engine will now flush any recently deleted objects after each command as long as they were created during the current event handling loop.

If an object is created during one event handling loop, and then deleted during another nested event handling loop it won't be flushed until control returns to the original event handling loop.

The upshot is that in tight loops, creating and deleting objects will result in objects being flushed immediately, reducing memory usage and making it easier to right object processing code which creates and deleted many objects.
Ability to set the dontUseQT property for a player object (windows only) (6.7.6-rc-1)

It is now possible to set the dontUseQT property for a player object. On Windows, the default value of the dontUseQt (global) property is false. This means that any player object created will use the QuickTime API for multimedia playback. With this new feature, you can set the dontUseQT property of a player to true, without changing the value of the global dontUseQt property. In that way you can have both QuickTime and non-QuickTime players playing at the same time.

Default buttons lose their glow when another button is clicked on Yosemite. (6.7.6-rc-1)

On Yosemite, default buttons should always glow regardless of whether other buttons are pressed (unless the app is in the background).

Font mapping feature (6.7.5-rc-1)

From 6.7.5 RC 1, a support for font name mapping had been added in the standalone builder settings for iOS. Updating to iOS 8.1 caused some stacks to startup extremely slowly, and that is due to the fact that setting a font using a name different from its actual PostScript name has gotten dramatically slower from iOS 8.1 (that slowdown was already present earlier, but not that noticeable).

We internally added a mapping file for all the default iOS fonts, so that the displayed names will not change.

As for the custom fonts any user would add in an application, we are not able to provide such a mapping file. Whence adding a custom font mapping file to a project is now possible from the iOS standalone settings.

Font mapping file

Each font name is mapped to its PostScript name in this way, one mapping per line:

<font name>=<PostScript name> <font name 2>=<PostScript name 2>

- font name is the name used in the call set the font of <object> to <font name>
- PostScript Name is the PostScript name as referenced from the Font Book.

Relocation of resources for Mac standalone applications (6.7.5-rc-1)

As stated in their Technical Note 2206, Apple does not allow application to have unsigned code in the MacOS folder.

To comply with this new rule, every non-executable file is now copied in the folder <App>.app/Contents/Resources/_MacOS.

As for the executable files, they are still copied alongside the engine, and located in specialFolderPath("engine").

This change does not impact any of the engine functions, which automatically redirect any path targeting the former location of the resource files.

However, the same does not apply to the externals, as they cannot determine if a path belongs to the app bundle or not; that affects also LiveCode external, such as revDatabase, this relocation breaks any script calling externals with a resource file.
In order to help with this overcoming this issue, the option Resources has been added for specialFolderPath on Desktop platforms, in LiveCode 6.7.2.

That will allow you to get the right path to the resources, being in development or standalone mode.

For instance, a code which used to be

```plaintext
// Get the place Resources folder
put the filename of this stack into tDatabasePath
delete the last item of tDatabasePath
// Append the SQLite file
put "/database/db.sqlite" after tDatabasePath
put revOpenDatabase("sqlite",tDatabasePath) into tDatabaseID
```

can be updated to

```plaintext
put specialFolderPath("resources") into tDbPath
// Append the SQLite file
put "/database/db.sqlite" after tDatabasePath
put revOpenDatabase("sqlite",tDatabasePath) into tDatabaseID
```

Add revBrowser error callback messages. (6.7.4-rc-1)

Added two new callbacks sent by revBrowser.
Note that these messages are only sent from browsers opened with revBrowserOpenCEF.

Callback messages:

- browserDocumentFailed
  - sent to the current card when the browser has encountered an error while loading a URL.
- browserDocumentFailedFrame
  - sent to the current card when the browser has encountered an error while loading a URL into a frame.

New iOS status bar style (6.7.4-rc-1)

What has changed?

The way the status bar is displayed changed in iOS 7. Previously, if the status bar was visible, the app view was shifted down by height of the status bar (20 pixels). From iOS 7 and above, the app view is given these 20 pixels of extra height, and the status bar is displayed on top of the app view (i.e there is an overlapping). This is a feature added by Apple, to give developers control over what appears behind the content of the status bar.

Some users reported this change of behaviour as a bug, since it changed the way their existing stacks were displayed in iOS 7 and above. A quick fix to this would be updating the engine to move the app view down 20 pixels, if the status bar is opaque. However, this fix would not guarantee backwards compatibility, since some users may have already adjusted for iOS 7 status bar behaviour, and have modified their code to work with those changes.

So we decided to add a new solid status bar style, which is opaque and automatically shifts down the view content by 20 pixels. So the difference between åœœopaqueåœ€ and åœœsolidåœ€ status bar styles is the following:

```plaintext
| pre-iOS 7 | iOS 7+
```
opaque | move the stack below the status bar | the status bar will be over the top
solid | move the stack below the status bar | move the stack below the status bar

[[ In App Purchase ]] Calling mobileStoreRestorePurchases when there are no previous purchases to restore (6.7.4–rc–1)

What has changed

Previously, if mobileStoreRestorePurchases was called and no previous purchases were made with that user account, nothing happened.
Now, a purchaseStateUpdate message is sent with state=restored and productID=""

Voice Over support (6.7.1–rc–4)

What has changed?

Prior to LC 6.6.4 RC3 one could not interact with LC stacks when Voice Over was turned on, for example, a LC button would not respond to touch events. This changed in LC 6.6.4 RC3, but broke interaction with native iOS controls, for example, links inside a mobile browser window did not work as expected with Voice Over. To address this issue, we added a new property to iOS native controls, **ignoreVoiceOverSensitivity**, that allow turning off Voice Over sensitivity for those objects.

Example:

```livecode
on ShowVOBrowser
  global gVOBrowserID
  if the platform = "iPhone" then
    if gVOBrowserID = empty then
      MobileControlCreate "browser"
      put the result into gVOBrowserID
    end if
    mobilecontrolset gVOBrowserID,"dataDetectorTypes","link"
    mobilecontrolset gVOBrowserID, "ignoreVoiceOverSensitivity","true"
    mobilecontrolset gVOBrowserID,"rect",0,0,300,300
    mobilecontrolset gVOBrowserID,"visible",true
  end if
end ShowVOBrowser

on hideVOBrowser
  global gVOBrowserID
  if the platform = "iPhone" then
    mobilecontrolset gVOBrowserID,"visible",false
    mobilecontrolset gVOBrowserID, "ignoreVoiceOverSensitivity","false"
  end if
```

Moreover, we added a new LiveCode function, `mobileIsVoiceOverRunning`, to detect if Voice Over is turned on.

Cocoa Support (6.7.0)

With 6.7 we have replaced the majority of Carbon API usage with Cocoa. The goals of this work are three-fold:

- Allow embedding of native 'NSViews' into LiveCode windows (in particular, browser controls).
- Enable submission of LiveCode apps to the Mac AppStore.
- Enable eventual building of 64-bit versions of LiveCode for Mac.

We have achieved the first two of these goals in 6.7.

The instability issues caused by the AppStore sandbox when using mixed Cocoa and Carbon APIs has been resolved - LiveCode apps built with 6.7 can be successfully sandboxed and thus submitted to the AppStore.

The `dontUseQT` property is now true by default on Mac. This means that, by default, the AVKit implementation of the player will be used on 10.8 and above. Note that, as it stands, when `dontUseQT` is true neither QT visual effects nor sound recording will work.

The final goal (64-bit support) will be gradually worked towards over the next few LiveCode versions as the engine gets 'decarbonated' (usage of Carbon APIs which do not have 64-bit equivalents removed).

An important internal change which will affect maintainers of Mac externals that use the `windowId` is that this property now returns the 'global window number' (which is the unique ID the Window Server uses to identify windows). To turn this into a Cocoa NSWindow pointer use `[NSApp windowWithWindowNumber: t_window_id]`. Note that it is no longer possible to get a Carbon WindowRef, nor should this be attempted as trying to mix Carbon and Cocoa in this manner will cause instability inside the sandbox environment required by the Mac AppStore.

An important script visible change that has occurred due to the move to Cocoa is screen updating. Previously (when using Carbon) the OS would 'coalesce' successive requests to update the screen - the window buffer would be updated, but the window buffer would only be flushed when the OS decided to. In Cocoa, after a screen update the window buffer is always flushed. Outside of 'lock screen', the engine applies any screen updates after each command execution therefore in 6.7+ make sure you use lock screen around blocks of code that make many screen updates - unless you want each update to be visible. It should be noted that the behavior in 6.7 is now the same as on Windows and Linux however the OS takes longer to flush window updates to the screen on Mac than on the other platforms meaning that using lock screen is important.

Note: QTVR movies are no longer supported as they are not supported by QTKit nor AVKit.

Note: Drawers no longer work on Mac, they will appear as normal stacks.

Location Services Disabled with LC 6.6.4 (rc1) (6.7.0)

A new function `mobileLocationAuthorizationStatus` (or `iphoneLocationAuthorizationStatus`) has been added. This returns the current location authorization status of the calling application. The status can be one of the following:

- **notDetermined**: User has not yet made a choice with regards to this application
- **restricted**: The application is not authorized to use location service
- **denied**: User has explicitly denied authorization for this application, or location services are disabled in...
Settings.
- **authorizedAlways**: User has granted authorization to use their location at any time, including monitoring for regions, visits, or significant location changes.
- **authorizedWhenInUse**: User has granted authorization to use their location only when the app is visible to them (it will be made visible to them if you continue to receive location updates while in the background). Authorization to use launch APIs has not been granted.

We have also changed the flow of the messages being sent to the user when using Location Services in iOS 8:

- In the standalone application settings tab, the developer can choose the type of the authorization request for their app. The two available options are either "always" or "when in use". Selecting "always" means that the app will prompt the user to grant authorization to use their location at any time, including monitoring for regions, visits, or significant location changes. The app then has access to the user's location even when the app is in the background. On the contrary, if "when in use" is selected, the app will prompt the user to grant authorization to use their location only when the app is visible on screen. You can choose only one type, not both. This means that if you go to Settings -> Privacy -> Location, you will see only two choices available ("Never" and either "Always" or "While using the app") for this app, keeping it consistent with other iOS apps.

- When the app is installed (on device or simulator) for the very first time, a dialog will pop up asking the user to authorize the app to use their location "always" or "when in use", depending on what was previously chosen in the standalone application settings.

- Every time the app is launched, it remembers the user's preference. No other popup dialogs will appear.

- The user can at any time change their preferences in Settings -> Privacy -> Location -> ..

- In that way, you need not modify your existing scripts that used Location Services, in order to add iOS 8 support.

**iOS 8 Support (6.7.0-rc-3)**

Support for iOS 8 device and simulator builds has been added to 6.6.4-rc-1 for OS 10.9 users. This means that if you are using OS 10.9 you must now have Xcode 6 installed in order to perform device builds. The requirements for all previous OS X versions will remain the same.

Bugs relating to orientation, push notifications and screen sizes on iOS 8 have been resolved in addition to standalone builder updates allowing for the specification of new iPhone 6 splash screens.

**Copy files do not work with the iOS 8 simulator (6.7.0-rc-3)**

This fix has been tweaked for 6.6.4-rc-3. If, when attempting to deploy to the iOS 8 simulator you get the error "Unable to start simulation: Unable to run app in Simulator", delete any previous version of the app installed on the simulator and redeploy.

**Non-executable file redirection on Mac (6.7.0-rc-2)**

Mac AppStore rules require that only executables (including bundles and apps) are present within the Contents/MacOS folder in the application bundle. However, historically (for cross-platform purposes), LiveCode applications traditional place resources relative to the engine executable, resulting in non-executable files to be present in the Contents/MacOS folder which violates AppStore signing policy. To remedy this situation without requiring users to change scripts, a simple redirection facility has been
implemented in the engine:
If an attempt is made to open a file for read which falls within Contents/MacOS and does not exist, the
engine will attempt to open the same path but under Contents/Resources/_MacOS instead.
If an attempt is made to list files in a folder which falls within Contents/MacOS, the engine will list files in
that folder and concatenate them will files within the same folder under Contents/Resources/_MacOS.
Additionally the standalone builder has had an extra processing step added on Mac:
After the Mac bundle has been built, the S/B recurses through Contents/MacOS and creates an identical
folder structure based at Contents/Resources/_MacOS. All non-executable files in any folders under
Contents/MacOS are moved to the same folder under Contents/Resources/_MacOS whereas any Mach-O
executable files are left where they are.
The result of this is that after building a standalone, from a script's point of view nothing has changed; but
the app bundle will conform to the rules required for signing for the Mac AppStore.

Determining the edition of the running engine (6.7.0–dp–9)

There is a new global property the editionType. This property can be used to determine what edition of the
engine is currently running.
It returns community if the currently running engine is the LiveCode Community IDE or a standalone built
with LiveCode Community.
It returns commercial if the currently running engine is the LiveCode Commercial IDE or a standalone built
with LiveCode Commercial.

Sound recording returns error message (6.7.0–dp–9)

The Quicktime implementation of sound recording has been updated to use the SGAudioMediaType in place
of the deprecated SoundMediaType, to fix various errors with sound recording. Moreover a pausing feature
has been added to the sound recorder, via the syntax record pause and record resume.

Threaded Rendering (6.7.0–dp–8)

In an effort to boost graphic rendering performance, when LiveCode is run on a machine with multiple cores,
all stacks with accelerated rendering turned off will be split into multiple tiles, with each tile being rendered
individually (and ideally concurrently). The number of tiles the stack is split into depends upon the machine
the stack is running on.

So, for example, on a machine with 4 or more cores, a stack will be split up into 4 rectangular tiles - top left,
top right, bottom left, bottom right. (On dual core machines, we use a top half bottom half split). Each tile
will be rendered individually, meaning that in an ideal situation, we have 4 concurrent drawing operations
providing a 4 fold speed increase. This is the ideal, but in reality, in the above case, we see closer to a 2
fold improvement. This will obviously vary with the number of concurrent operations the given machine can
perform.

Note that this feature is currently disabled on Windows due to threading conflicts caused by the
player object.

Effective points of graphics (6.7.0–dp–7)

You can now use the effective points' and 'the effective relativePoints' properties of a graphic object to fetch
a polygon representation of rectangle, round rectangle, oval and regular polygon graphic objects.

Player messages aren't sent correctly. (6.7.0–dp–7)

The occurrence of playStarted, playPaused and playStopped messages has been cleaned up.
The playStarted message will only be sent when the rate of the movie changes from zero to non-zero - whether via clicking the play button, setting the playRate or by using play start / play resume via script.

The playPaused message will only be sent when the rate of the movie changes from non-zero to zero - whether via clicking the pause button, setting the playRate or by using play pause / play stop.

The playStopped message will only be sent when the movie reaches the end of playback.

These are the only cases in which the messages will be sent - in particular, setting the filename will no longer send any messages and you will not get multiple messages of the same type in succession.

Showing a modal dialog confuses mouse state. (6.7.0-dp-6)

When a modal dialog is shown, the engine will now immediate reset the mouse state to ensure the context of events is the new dialog. In particular, mouseRelease will be sent if the mouse is down and mouseLeave will be sent if the mouse is within the previous window.

Nine-way stretch for images (6.7.0-dp-5)

You can now set 'the centerRect' property of an image. This property should be a rectangle, with coordinates relative to the formattedRect of the image. The property specifies the area of the image that should be stretched when the image is scaled. For example, if the centerRect of an image which is 16x16 is set to 4,4,12,12 then:

- The 4x4 corner portions of the image will not be stretched
- The top 4x8 and 8x4 side portions of the image will be stretched horizontally or vertically (depending on orientation)
- The middle 8x8 area will stretch to fill the middle.

This property is useful for using images as backgrounds to buttons and interface elements - allowing a non-stretched border with stretched interior to be specified.

Updated text rendering for iOS and OS X (6.7.0-dp-5)

In order to improve performance, the text rendering routines for iOS and OS X and been updated to use the latest APIs. This has had a significant improvement in the text rendering speed, particularly on OS X.

It's worth noting that the previous OS X routines used synthesised font styles. That is, bold and italic styles were emulated (by slanting or thickening) if the font being rendered was not bold or italic. This is not the case for the new routines. If there is not a font present on the system with the given style, the plain alternative will be used.

The is the case with the default font - Lucida Grande. By default, systems only come with the bold variant. Thus, if you have a field with the default font and italic style, it will be rendered in plain style.

As part of LiveCodes progression toward unicode, the use of older symbol fonts is no longer fully supported. In order to ensure symbol fonts are drawn correctly, the font must be unicode encoded. The fonts that come with the latest versions of OS X are all unicode encoded.

Export snapshot with metadata (6.7.0-dp-5)

An optional with metadata <metadata array> clause has been added to the export snapshot command. Currently the only metadata key that is implemented is density which can be used to include pixel density metadata in pixels per inch.
For example:

```
put 144 into theMetadataA["density"]
```

export snapshot of group 1 at size the width of group 1 2, the height of group 1 2 with metadata
theMetadataA

New variant of open and secure socket (6.7.0–dp–5)

New variants of open and secure socket have been added:

```
open secure socket socket with verification for host host
secure socket socket with verification for host host
```

The new host parameter allows the user to specify the host name the connection should be verified against.
This is particularly useful if server your socket is directly connected to is not the end host you are talking to.
For example when tunnelling through a proxy to connect to a HTTPS URL.

Multiple density image support for patterns. (6.7.0–dp–5)

This extends the existing image resolution independence features to any pattern using that image.
When an object pattern is set to a multiple density image, that pattern will automatically use the best source image for the density at which it is drawn.

QT-related features don't work. (6.7.0–dp–5)

QT effects and sound recording will now work as long as 'dontUseQT' is set to false. In this case, the player will default to using QTKit.
If you are submitting an app to the Mac AppStore, or wish to use AVFoundation player on 10.8 and above, ensure that dontUseQT is set to true in your startup handler, or before any code or stack which uses QT is run.

Objects which are adjacent don't necessary appear so at non integral scale factors. (6.7.0–dp–5)

At non-integral scale factors (such as 150% Hi-DPI mode on Windows), objects which should appear next to each other can have a visible channel.
This issue isn't completely fixable due to the nature of approximations used when compositing to the screen. However, this problem has been mitigated in a couple of ways - firstly antialiasing is forced on whenever the scale factor is non-integral; secondly clipping rectangles always fall on device pixel boundaries.

Queuing too many pending messages causes slowdown and random crashes. (6.7.0–dp–5)

A limit on the number of user-defined pending messages (those created with 'send in time') has been imposed. If there are more than 64k messages in the pending message queue, 'send in time' will now throw an error when attempting to queue another one.
This limit has been imposed to prevent engine lock up and eventual instability due to memory exhaustion in the case that pending message loops cause rapid increases in the number of pending messages.
Inconsistencies in behavior when doing 'delete the selectedChunk'. (6.7.0–dp–5)

The following should all operate the same way after selecting a line in a field by doing 'triple-click', or just selected the whole line without the paragraph break:

- pressing backspace
- executing 'delete the selectedChunk'
- executing 'get the selectedChunk; delete it'

Previously, 'delete the selectedChunk' would cause paragraph styles not to be set correctly on the resulting paragraph; or the paragraph break to be included when it should not be - this is no longer the case.
Previously, 'get / delete it' would only work correctly the first time the command was executed - this is no longer the case.

In-App Purchasing (6.7.0–dp–3)

Why has the API changed?

The LiveCode engine until now supported in-app purchasing for apps distributed through the Google Play store (formerly Android Market), as well as the Apple AppStore. This support is now extended so that apps distributed through other avenues (the Amazon & Samsung app stores) can make use of the in-app purchase features provided. For this reason, new LiveCode commands have been added, and some of the old ones have slightly changed. However, all of the old commands are still supported (for the Google Play Store and the Apple AppStore). In order the existing scripts users have written to continue to work, all it needs is to add one or two extra lines, depending on the store. More details on this later. Moreover, the new API allows the user to query specific product information (such as price, description etc) before they make a purchase, and supports purchasing of subscription items for all available stores. Furthermore, for the Google Play Store, the new API uses the newest version of Google In-App Billing API (v3), that offers synchronous purchase flow, and purchase information is available immediately after it completes. This information of in-app purchases is maintained within the Google Play system until the purchase is consumed. More on the consumption of purchased items later.

Note that our in-app purchasing implementation for the Amazon Appstore is unfortunately restricted to commercial license holders only. This is due to the Amazon PML license being incompatible with the GPL. If you wish to use the Amazon in-app purchasing features of LiveCode you will need to be a valid commercial licence holder.

What has changed?

To start with, the main changes are the following:

- Each item has an extra property, the `itemType`, that has to be specified before making a purchase. This is done using the `mobileStoreSetProductType` command. The `itemType` can either be `subs`, for subscription items, or `inapp` for consumable and non-consumable items.

- Due to a restriction of the newest version of Google In-App Billing API, you cannot buy consumable items more than once, unless you consume them. This is done using the `mobileStoreConsumePurchase` command. Note that this command is actually only used when interacting with the Google Play Store API. What it does is sending a consumption request to Google Play, so that you will be able to buy this product again. You would typically implement consumption for items that can be purchased multiple times (i.e. for consumable products, such as in-game currency, fuel etc). Note that in case you call `mobileStoreConsumePurchase` on a non consumable product, then you no longer own this item.

- The new purchase flow has become simpler.
Instead of
- creating a purchase request (mobilePurchaseCreate productID)
- store the new purchase request ID (put the result into tPurchaseID),
- setting properties such as quantity and developer payload (mobilePurchaseSet tPurchaseID, "quantity", pQuantity)
- sending a purchase request to the store (mobilePurchaseSendRequest tPurchaseID)
now all it needs is just
- set the product type (mobileStoreSetProductType productID, itemType)
- make a purchase (mobileStoreMakePurchase productID, quantity, developerPayload)

- The purchaseStateUpdate message that the store sends in response to
  mobileStoreMakePurchase, contains not only the purchase identifier and the state of the purchase,
  but also the product identifier of the requested item:

purchaseStateUpdate purchaseID, productID, state

- So you can query a purchased product property using the product identifier, instead of the purchase
  identifier:

mobileStoreProductProperty productID, propertyName

Note that the old function mobilePurchaseGet purchaseID, propertyName will still work.

- You can get information on a specific item (such as product identifier, product type, price etc), using
  the mobileStoreRequestProductDetails command. The store responds:

In case the request is successful, a productDetailsReceived message is sent by the store.

In case of failure, a productRequestError message is sent by the store.

- You can get a list of all known completed purchases using mobileStorePurchasedProducts
  function. This returns a list of product identifiers of restored or newly bought purchases.

What needs to change in existing scripts?

It is recommended that scripts which were written using previous versions of LiveCode (and thus use the old
LiveCode API for in-app purchasing), should be used to run on these versions. However, it is still possible to
run an existing script (that makes use of in-app purchasing feature) on LiveCode 6.7, only by changing a few
things:

- purchaseStateUpdate message is now called with 3 parameters, (purchaseID, productID, state), instead
  of two (purchaseID, state). This applies to apps built for both the Google Play Store and the Apple AppStore.

- before sending a mobilePurchaseSendRequest, you have to specify the type (subs or inapp) of the item
  using mobileStoreSetProductType productID, type command (Google Play Store only).

- if you want to buy more than one consumable item, you have to consume it first. This can be done by
  using the mobileStoreConsumePurchase productID command (Google Play Store only).

If you want to build apps for Amazon and/or Samsung Store, you have to use the newest LiveCode API. Â
**How to use the new API?**

**Setup**

Before you can use IAP, you must set up products in each vendor's developer portal. In brief, you have to:

- Create each product you want to sell, giving it a unique identifier. Note that for the Samsung Seller Office, the developer cannot choose the product identifier. This is assigned by the store.
- Submit the items for approval to the appropriate store. Some stores may require additional metadata, such as screenshots of your for sale items.
- Set up unique test accounts. The user is not charged when making a purchase using the test account details. This applies to Apple and Google. Amazon and Samsung have different methods for testing.

For more detailed store-specific information, you can have a look at the links below:

Apple AppStore

Google Play Store

Amazon Appstore

Samsung Apps Store and more specifically click [here](#)

**Purchase Types**

There are three classes of products users can purchase:

1. One-time purchases that get "consumed". Typically, these items are called **consumables**. The user can buy as many times as they want (virtual coins/bullets in a game), except in apps built for the Google Play Store, where the user has to consume the purchased item first, and then buy (one) more.
2. One-time purchases that last forever, such as unlocking extra features, downloading new content once. These items are usually called **non-consumables**.
3. Subscriptions where the app user pays a periodical fee to receive some ongoing service. Subscriptions can either be auto-renewable or non-renewable.

Each vendor uses different terminology for these purchases:

<table>
<thead>
<tr>
<th></th>
<th>Apple</th>
<th>Google</th>
<th>Amazon</th>
<th>Samsung</th>
</tr>
</thead>
<tbody>
<tr>
<td>one-time, gets consumed</td>
<td>consumable</td>
<td>unmanaged</td>
<td>consumable</td>
<td>consumable</td>
</tr>
<tr>
<td>one-time, lasts forever</td>
<td>non-consumable</td>
<td>managed</td>
<td>entitlement</td>
<td>non-consumable</td>
</tr>
<tr>
<td>subscriptions</td>
<td>auto-renewable, non-renewable</td>
<td>auto-renewable</td>
<td>auto-renewable</td>
<td>non-renewable</td>
</tr>
</tbody>
</table>

**Testing**

Again, each store uses a different method of testing.

For the Apple AppStore, you can create test accounts. More details [here](#).

For the Google Play Store, you can create test accounts as well as test using static responses. More details [here](#). Note that you cannot test subscriptions using the test account. This means that the test user will be charged when purchasing a subscription item. A possible workaround to this, is to log into the Google Wallet Service as a seller, using your Google Developer account details, and "refund" and then "cancel" the
order of the subscription item that the test user had just purchased.

For the Amazon Appstore, you can test your app using SDK Tester. This is a developer tool that allows users of the Amazon Mobile App SDK to test their implementation in a production-like environment before submitting it to Amazon for publication. More details [here](#).

For the Samsung Apps Store, Samsung IAP API offers three modes to test the service under various conditions: **Production Mode**, **Test Mode Success**, **Test Mode Fail**. During development period, you can select the mode in the Standalone Application Settings window. Before releasing your application, you must change to Production Mode. If you release your application in Test Mode, actual payments will not occur. More details on page 6 and 7 [here](#).

Note that in Production Mode, your app can only interact with item groups with sales status. This information exists in the Samsung Seller Office. However, item groups are only given sales status after the app has been certified. In other words, you can test your app in Production Mode only after it has been certified by Samsung.

**Syntax**

Implementing in-app purchasing requires two way communication between your LiveCode app and the vendor's store. Here is the basic process:

- Your app sends a request to purchase a specific in-app purchase to the store
- The store verifies this and attempts to take payment
- If payment is successful the store notifies your app
- Your app unlocks features or downloads new content / fulfils the in-app purchase
- Your app tells the store that all actions associated with the purchase have been completed
- Store logs that in-app purchase has been completed

**Commands, Functions and Messages**

To determine if in-app purchasing is available use:

`mobileStoreCanMakePurchase()`

Returns `true` if in-app purchases can be made, `false` if not. Throughout the purchase process, the store sends `purchaseStateUpdate` messages to your app which report any changes in the status of active purchases. The receipt of these messages can be switched on and off using:

`mobileStoreEnablePurchaseUpdates`
`mobileStoreDisablePurchaseUpdates`

If you want to get information on a specific item (such as product identifier, product type, price etc), you can use:

`mobileStoreRequestProductDetails productID`

The `productID` is the identifier of the item you are interested. Then, the store sends a `productDetailsReceived` message, in case the request is successful, otherwise it sends a `productRequestError` message:

`productDetailsReceived productID, details`

The `productID` is the identifier of the item, and `details` is an array with the following keys - that are different depending on the store:
For Android stores (Google, Amazon, Samsung), the keys are:

- **productID** : identifier of the requested product
- **price** : price of the requested product
- **description** : description of the requested product
- **title** : title of the requested product
- **itemType** : type of the requested product
- **itemImageUrl** : URL where the image (if any) of the requested product is stored
- **itemDownloadUrl** : URL to download the requested product
- **subscriptionDurationUnit** : subscription duration unit of the requested product
- **subscriptionDurationMultiplier** : subscription duration multiplier of the requested product

Note that some Android stores do not provide values for all the above keys. In this case, the value for the corresponding key will be empty.

For iTunes Connect store (Apple), the keys of `details` array are the following:

- **price** : price of the requested product
- **description** : description of the requested product
- **title** : title of the requested product
- **currency code** : price currency code of the requested product
- **currency symbol** : currency symbol of the requested product
- **unicode description** : unicode description of the requested product
- **unicode title** : unicode title of the requested product
- **unicode currency symbol** : unicode currency symbol of the requested product

If `mobileStoreRequestProductDetails` is not successful, then a `productRequestError` message is sent:

```plaintext
productRequestError productID, error
```

The `productID` is the identifier of the item, and `error` is a string that describes the error.

Before sending a purchase request for a particular item, you have to specify the type of this item. To do this, use:

```plaintext
mobileStoreSetProductType itemType
```

The `itemType` can either be `subs` or `inapp`.

To create and send a request for a new purchase use:

```plaintext
mobileStoreMakePurchase productID, quantity, developerPayload
```

The `productID` is the identifier of the in-app purchase you created in the vendor's developer portal and wish to purchase. The `quantity` specifies the quantity of the in-app purchase to buy (iOS only - always "1" in Android). The `developerPayload` is a string of less than 256 characters that will be returned with the purchase details once complete. Can be used to later identify a purchase response to a specific request (Android only).

To get a list of all known completed purchases use:

```plaintext
mobileStorePurchasedProducts()
```

It returns a return-separated list of product identifiers, of restored or newly bought purchases which are confirmed as complete. Note that in iOS, consumable products as well as non-renewable subscriptions will not be contained in this list.
Once a purchase is complete, you can retrieve the properties of the purchased product, using:

**mobileStoreProductProperty** (productID, property)

The parameters are as follows:

- **productID** : identifier of the requested product
- **property** : name of the purchase request property to get

Properties which can be queried can differ depending on the store:

For the Samsung Apps Store (Android), you can query the properties:

- **title** : title of the purchased product
- **productId** : identifier of the purchased product
- **price** : price of the purchased product
- **currencyUnit** : currency unit of the product price
- **description** : description of the product as specified in the Samsung Seller Office
- **itemImageUrl** : URL where the image of the purchased product is stored
- **itemDownloadUrl** : URL to download the purchased product
- **paymentId** : payment identifier of the purchased product
- **purchasedId** : purchase identifier of the purchased product
- **purchaseDate** : purchase date, in milliseconds
- **verifyUrl** : IAP server URL for checking if the purchase is valid for the IAP server, using the purchasedId value

For the Google Play Store (Android), you can query the properties:

- **productId** : identifier of the purchased product
- **packageName** : application package from which the purchase originated
- **orderId** : unique order identifier for the transaction. This corresponds to the Google Wallet Order ID
- **purchaseTime** : time the product was purchased, in milliseconds
- **developerPayload** : developer-specified string that contains supplemental information about an order. You can specify a value for this in **mobileStoreMakePurchase**
- **purchaseToken** : token that uniquely identifies a purchase for a given item and user pair.
- **itemType** : type of the purchased item, inapp or subs
- **signature** : string containing the signature of the purchase data that was signed with the private key of the developer. The data signature uses the RSASSA-PKCS1-v1_5 scheme

For the Amazon Appstore (Android), you can query the properties:

- **productId** : identifier of the purchased product
- **itemType** : type of the purchased product. This can be **CONSUMABLE**, **ENTITLED** or **SUBSCRIPTION**
- **subscriptionPeriod** : string indicating the start and end date for subscription (for subscription products only)
- **purchaseToken** : purchase token that can be used from an external server to validate purchase

For Apple AppStore (iOS), you can query the properties:

- **quantity** : amount of item purchased. You can specify a value for this in **mobileStoreMakePurchase**
- **productId** : identifier of the purchased product
- **receipt** : block of data that can be used to confirm the purchase from a remote server with the iTunes Connect store
- **purchaseDate** : date the purchase / restoration request was sent
- **transactionIdentifier** : unique identifier for a successful purchase / restoration request
- **originalPurchaseDate** : date of the original purchase, for restored purchases
- **originalTransactionIdentifier** : the transaction identifier of the original purchase, for restored purchases
- originalReceipt : the receipt for the original purchase, for restored purchases

Once you have sent your purchase request and it has been confirmed, you can then unlock or download new content to fulfill the requirements of the in-app purchase. You must inform the store once you have completely fulfilled the purchase using:

**mobileStoreConfirmPurchase** *productID*

Here, *productID* is the identifier of the product requested for purchase.

**mobileStoreConfirmPurchase** should only be called on a purchase request in the *paymentReceived* or *restored* state (more on the states of the purchase later). If you don’t send this confirmation before the app is closed, **purchaseStateUpdate** messages for the purchase will be sent to your app the next time updates are enabled by calling the **mobileStoreEnablePurchaseUpdates** command.

To consume a purchased product use:

**mobileStoreConsumePurchase** *productID*

Here, *productID* is the identifier of the product requested for consumption. Note that this command is actually only used when interacting with the Google Play Store API. This is because the Google Play Store API has a restriction that ensures a consumable product is consumed before another instance is purchased. *Consume* means that the purchase is removed from the user’s inventory of purchased items, allowing the user buy that product again.

Note that **mobileStoreConsumePurchase** must only be called on consumable products. If you call **mobileStoreConsumePurchase** on a non-consumable product, then you no longer own this product.

To instruct the store to re-send notifications of previously completed purchases use:

**mobileStoreRestorePurchases**

This would typically be called the first time an app is run after installation on a new device to restore any items bought through the app.

To get more detailed information about errors in the purchase request use:

**mobileStorePurchaseError** *(purchaseID)*

The store sends **purchaseStateUpdate** messages to notifies your app of any changes in state to the purchase request. These messages continue until you notify the store that the purchase is complete or it is cancelled.

**purchaseStateUpdate** *purchaseID, productID, state*

The state can be any one of the following:

- **sendingRequest** : the purchase request is being sent to the store / marketplace
- **paymentReceived** : the requested item has been paid for. The item should now be delivered to the user and confirmed via the **mobileStoreConfirmPurchase** command
- **alreadyEntitled** : the requested item is already owned, and cannot be purchased again
- **invalidSKU** : the requested item does not exist in the store listing
- **complete** : the purchase has now been paid for and delivered
- **restored** : the purchase has been restored after a call to **mobileStoreRestorePurchases**. The purchase should now be delivered to the user and confirmed via the **mobileStoreConfirmPurchase** command
- **cancelled** : the purchase was cancelled by the user before payment was received
- **error** : An error occurred during the payment request. More detailed information is available from the
mobileStorePurchaseError function

Clipboard data 'styledText' array accessor. (6.7.0–dp–1)

A new clipboard format has been added 'styledText'. This format returns (or sets) the clipboard to a styled text array - the same format as the 'styledText' property of field chunks. All text formats can convert to and from the 'styledText' key.
For example, you can now do:

- set the clipboardData["styledText"] to the styledText of line 5 of field 3
- set the styledText of line 6 of field 3 to the clipboardData["styledText"]

Note that the dragData can now also be used with this new format in exactly the same way.

Improved revBrowser external (6.7.0–dp–1)

The revBrowser external has been updated to support Cocoa on OSX, and now embeds the browser control properly within the window.
In addition a new browser component based on CEF (Chromium Embedded Framework) has been added.
This new browser allows for a consistent appearance across all platforms with a modern, well supported feature set.

To use the new CEF browser use the revBrowserOpenCef command in place of revBrowserOpen. This will create a CEF browser instance which can be used with the existing revBrowser commands and functions in exactly the same way as before.

JavaScript integration

The new chrome browser allows us to add the ability to call LiveCode handlers from within the browser using JavaScript. To make a LiveCode handler visible to JavaScript, use the revBrowserAddJavaScriptHandler command, and to remove it use the revBrowserRemoveJavaScriptHandler command. LiveCode handlers are added as functions with the same name attached to a global 'liveCode' object. When called, these functions will result in the corresponding LiveCode handler message being sent to the browser card with the browser instance ID and any function arguments as parameters.

Example:

With the handler "myJSHandler" registered using revBrowserAddJavaScriptHandler, it can be called from the browser like so:

liveCode.myJSHandler(tFieldContents, tAction);

the LiveCode handler would then be called with the following parameters:

- pBrowserInstance (the browser instance id, as returned from the revOpenBrowserCef function)
- pFieldContents (the first argument of the JavaScript function call)
- pAction (the second argument of the JavaScript function call)

OS 10.5 (Leopard) Support (6.7.0–dp–1)

As of version 6.7-dp-1, Mac OS 10.5 (Leopard) support has been dropped from LiveCode. This is primarily for technical reasons: In order to support the latest OS X features (e.g. Cocoa) dropping 10.5 support was required.
As Leopard was the last Mac version to support PPC, support for the PPC architecture has also been dropped and the Universal and PPC options have been removed from the Standalone Builder.

Users wishing to produce 10.5 compatible executables can still do so using LiveCode version 6.6.x (and earlier).

Setting the label of an option or combo-box does not update the menuHistory. (6.7.0–dp–1)

Previously, setting the label of an option or combo-box control would not update the menuHistory property. Now, setting the label of such a control will search through the list of items in the control and set the menuHistory to the first item that matches (taking into account the setting of the caseSensitive local property).

Note: Unlike setting the menuHistory property direct, this does not cause a menuPick message to be sent.

Specific bug fixes (6.7.7–rc–4)
(bug fixes specific to the current build are highlighted in bold, reverted bug fixes are stricken through)

15898 Building 32-bit slice only causes app to crash on iPad 1

Specific bug fixes (6.7.7–rc–3)

15743 iOS standalone engine do not build anymore in Debug mode
15718 'SSL library not found' error thrown on iOS when using SSL & Encryption library
15705 strange flash occurring under iOS with visual effect push left/right
15703 iOS app saved with minimum version 8.4 won't install on a device running iOS 8.4
15685 Fix the path to the OSX standalone engine on Linux

Specific bug fixes (6.7.7–rc–2)

15654 LiveCode 6.7.7 RC 1 / 7.1.0 DP 1 won't install on 10.6.8
15642 revZip fails to open archive in 64-bit iOS
15625 "record sound" is not replacing existing sound file
15619 Opening sqlite on iOS return "revdberror"
15590 Crop image within a repeat loop causes crash to desktop after a few iterations
1569 Scrolling a group causes LiveCode when hiding scrollbar

Specific bug fixes (6.7.7–rc–1)

15572 French application menu is not translated.
15570 Do not run "after" handlers if an error occurs.
15566 Preferences Mobile Support can't find the SDK for Xcode 6.4
15561 Tooltip appears for controls out of the visible window
15511 Erroneous assertion failure logged on server when doing 'go stack'
15507 The filename / effective filename of the root script returns empty in LiveCode server.
15495 [Player] Progress Bar does not update when playing audio files and alwaysBuffer is true
15263 Large files can EOF prematurely on Linux.
15259 DataGrid library missing in standalones
14418 Player doesn't accept filenames starting with file://
14289 Update SQLite to 3.8.10.2
14056 Cannot hide cursor on Mac from LiveCode 6.7
13391  [[Player]] Make default foregroundColor a colour other than black
13390  [[Player]] foregroundColor and hiliteColor not saved with stack
13262  [[Player]] player object in hidden group still displays video image
12834  [[Player]] enhancement set colour of player controller buttons using backcolor property
11866  iphoneHeadingCalibrationTimeout returns empty
11803  set the RTFText of field does not work on Android
6791   Objects are only deleted on idle
5209   FormattedHeight does not take into account controller
4123   eraser tool in magnify palette is 1 pixel off

Specific bug fixes (6.7.6-rc-3)
  Make rawKeyUp and rawKeyDown consistent on non-Latin chars

Specific bug fixes (6.7.6-rc-2)
  15526   Trailing 'break point' in the points of a graphic not preserved.

Specific bug fixes (6.7.6-rc-1)
  15456   libURL handles Content-Transfer-Encoding incorrectly
  15432   Filename of stack not properly handling tilde character
  15416   Crash when building a SQLite query with out-of-bounds placeholders
  15408   Deleted objects can still be accessed in some circumstances.
  15401   the defaultFolder is initialised to the iOS SDK path
  15392   Setting the points of a graphic to a string with trailing returns causes erroneous points to be added.
  15385   keyUp on MacOS is sent with wrong value
  15372   Wait with messages causes crash on Mac OS
  15324   in iOS
  15292   Minimum Android Version - add 4.0
  15113   Default buttons lose their glow when another button is clicked on Yosemite.
  14684   2.2 Android Froyo standalone application crashes
  14642   IDE crashes after deleting a control
  14435   Problems with players
  14349   Can't use mouse events in QT players on a Mac
  9820   LiveCode Server provides no POST data when Content-Type is application/x-www-form-urlencoded; charset=<...>

Specific bug fixes (6.7.5-rc-3)
  15320   Standalones with externals quit when launched if built for multiple platforms
  15300   Cannot build iOS app using Xcode 6.2 on Yosemite
  15288   library dropbox working as standalone
  15244   revdb always put in standalones
  15232   Android - acceleration rendering causing flicker when turned on
  15110   Android turning on "acceleratedRendering" causes a black screen flash
  14257   openSSL version outdated
Specific bug fixes (6.7.5–rc–1)

All root folders are added (empty) in the Mac standalone's Externals folders if
~/Documents/My LiveCode/Runtime/Mac OS X/x86-32/Externals/externals.txt file does not exist

Xcode 6.3 cannot be selected for iOS deployment

go to card on mousedown causes crash on IDE

App quits with setting accelerated rendering on opening in iphone 5S and 6

set EnabledTracks does not actually set the enabled tracks

Hard crash on iOS during startup

Player becomes unresponsive when changing cards in a background

Android mobile app launch hangs up while processing the oncard handler

unable to test ios standalones on iPad 1 running ios 5.1.1

Mac OS too many screen updates when focussing field

read from socket x until empty returns nothing

mobileClearTouches clears touches after a mouseDown handler has completed

standalone built for Yosemite cannot connect to database if it is within the app bundle

acceleratedRendering property causes crash

Crash when using dictation

revVideoGrabber crashes in Windows 8

Fix a null-pointer crash on Linux server

Specific bug fixes (6.7.4)

mobilePick position wrong on iOS8.2 and iPhone 6 scaled

Specific bug fixes (6.7.4–rc–3)

Simulators listed in Preference > Mobile Support are not in the Development > Test Target list

Crash when sorting non-shared text field in a group

More Room For Prompt/Message in mobile busy indicator

Can't Use XCode Version 6.2 (6C131e)

[in-app purchase] Calling mobileStoreProductProperty(prodID

iOS fullscreen video controller not visible when tapping on movie rect

Using beep while audio is playing on mobile

Wrong Splash screen on iPhone 6+

Script Editor incorrectly highlights block comments as if they nest.

Specific bug fixes (6.7.4–rc–1)

Android: Receive a purchaseStateUpdate message with state=complete after calling mobileStoreConfirmPurchase

Quitting LiveCode with revBrowserCEF window open causes a hang

Option menu crashes app in Simulator 8.0

fatal exception - Service not registered: com.runrev.android.billing.google.IabHelper

iOS standalones with RevZip externals and iOS Min Version >= 7.0 do not build

Intermittent crash on MCPlatformWindowMaskRelease()

video controller not responding
IDE crashes when viewing with Inspector
MobilePick Fails on Android 5 every other load
fiddling with alphaData crashes/hangs LC 7.0.2
[[ In-app purchase ]] mobileStorePurchasedProducts returns empty on iOS
Google Play Store - No Information on Product Details
Google Play Store - Extra error message when an in-app purchase fails
[[ In App Purchase ]] Calling mobileStoreRestorePurchases when there are no previous purchases to restore
Location of modal stacks incorrect on OSX
Printing is inconsistent
LC and standalones use wrong version icon in Ubuntu Launcher
iOS 7 Status Bar Hidden

Specific bug fixes (6.7.2)
Binary submitted to App Store rejected due to minimum version mismatch

Specific bug fixes (6.7.2–rc–2)
Text breaking can lead to hang LiveCode with Unicode text
Unable to enter Chinese characters in script editor
New resource folder location prevent from using 'launch' in standalones
Standalone crashes after closing a palette window
-crash with illegal imagesource manipulation
Menu items which point to submenus are selectable
[[Player]] Audio files do not play in player object
Faulty implementation of location services plist file in Livecode 6.7
iOS 8 GPS chips should be used even though NA is selected in standalone settings
Android externals libraries are not copied in the apk on Linux
Crash when pressing backspace key after writing an accented character

Specific bug fixes (6.7.2–rc–1)
Answer color returns the wrong color on OS X after using magnifying glass
IconGravity does not work in windows
mobileCanMakePurchase() stops in Android
Play audioclip loops when followed by wait until the sound is done
[[ Player ]] For some videos
[[Player]] documentation does not include error result for setting filename
Video Player not managing relative paths
backgroundpattern of a field out of sync if formattedHeight > roughly 32768
mobileExportImageToAlbum don't work on Android

Specific bug fixes (6.7.1)
Buttons with no icon but an icongravity make LiveCode crash on load
[[Player]] retains previously open movie when invalid path used.
[[Player]] does not return correct result when setting player filename to invalid path or corrupt movie
14211 referencing the next line causes the IDE to crash
14210 LiveCode IDE crashes changing textColor of the button
14185 Command D has to be pressed twice to trigger "Don't Save" in stack save dialog when triggered by Quit
13860 MobileExportImageToAlbum fails to export a referenced image in iOS

Specific bug fixes (6.7.1–rc–4)
14176 Mac mouse down sound plays every other time
14152 Waitâ€¦ with messages locks text typing
14141 IDE Unresponsive after setting the fileName of a player object during a playStopped message
14067 numpad key codes changed [regression]
14066 Export empty ranges if they have attributes
14019 Color picker can cause IDE to lock up
13871 Allow host name to IP address resolution when only loopback interfaces are configured.
13253 mobileBusyIndicatorStop not working properly when multiple instances of busy indicators are present on android
13169 mobileUpdateContact cannot be dismissed
13168 mobileShowContact cannot be dismissed
12899 mobilePick with empty list inside of inputBeginEditing crashes app
12784 Selecting text in field with traversalOn set to false confuses selectedField and focusedObject
12726 Crash if you copy the tempate button
11688 Take margins into account for pageHeight and pageRanges field props

Specific bug fixes (6.7.1–rc–3)
14098 answer color with color not correctly working LC 6.7.1 RC2
14093 Printing crashes on MAC
14070 text color in menuButton does not invert
14035 server-based video not working in LiveCode 7 or 7.0.1 (rc1 or rc2)
14034 Non-explicit return values from functions needs documenting
14028 export snapshot doesn't work on invisible objects
13927 Local notifications not working in iOS 8 simulator

Specific bug fixes (6.7.1–rc–2)
13979 MobilePickContact always returns empty in iOS8
13974 within(object
13970 export snapshot not working as expected for video snapshot exports
13968 [[Player]] Looping does not work within selection
13957 The IDE crashes when saving a specific project
13950 audioClip looping cannot be stopped
13949 Snapshot on iOS 8.0 is offset wrongly if orientation is landscape
13940 An application with no menus can not be quit with the menu or key commands
13938 setting alphaData does not replace alphaData but somehow adds to alphaData from LC 6.7 on
13919 LiveCode server unable to find handlers in script only stack
dontuseqt value is ignored in player creation when reopening a recently closed stack that contains a player
Standards menus can't be translated
command-alt-tab does not toggle browse/edit mode
Fix a default button highlight issue for Yosemite
Deleting whitespace at beginning of a script is buggy
Missing dictionary entry for "status" property
Setting defaultMenubar is flaky
Images with resizeQuality set to "good" are rendered oddly depending on coordinates
Corrupted stack file crashes LiveCode
External tail calls do not execute correctly on Desktop.
arrowKeys trigger mouse down in menu bar
Unable to uncomment lines of code via keyboard shortcut in OSX
Script Editor auto-indenting no longer works as in previous versions
Cards don't resize & redraw in wrong place after clicking the title bar
Sheet dialog won't stay at the right place
command "play stop" does not work
Fix the colour picker on OSX Yosemite

Specific bug fixes (6.7.1–rc–1)

13848  Popup menu items with Quit / Preferences or About are hidden on Mac.
13847  Cmd+
13842  On OSX Yosemite
13829  Change in how formattedRect of chunk is reported
13800  Could not generate package manifest when newer Android Build tools are installed
13799  Use Helvetica Neue as the default font on OSX Yosemite
13773  Make sure the first callback is always sent
13598  Instability when deleting a space before a handler definition on the very first line of a script.
13580  Export snapshot from rect at size selects the wrong source rectangle
12558  Older versions of LiveCode report a corrupt stack when opening a file from newer version
11912  On OSX
11647  After Android hardware backKey is pressed

Specific bug fixes (6.7.0)

13753  Project Browser reports incorrect control for behavior of a card
13752  Double-Clicking On a Player Doesn't Show Inspector
13750  Picker broken on iPhone 4 iOS 7.1
13746  the shape property of stacks is broken
13721  Externals using 'LCObjectPost' don't always cause the action to trigger - particularly on Yosemite.
13711  Player plays audio but not video
13710  [[Player]] video image not shown under some circumstances
13708  mobilepickphoto in landscape orientation causes an orientation change
13707  [[iOS 8]] Denying access to location services when the app is launched for the very first time causes the app to freeze
13699  iOS 8 Keyboard is invisible if privacy set to "While Using the App"
13677  iOS Picker appears under the keyboard on iOS 8
Scrollbar for the font selection in the script editor preferences doesn’t work

Ask/answer calls in (pre)openstack cause iOS 8 apps to hang

Make sure PATH variable passes through to shell() properly on Yosemite.

Location Services Disabled with LC 6.6.4 (rc1)

Shutdown request message sent twice when triggered from quit in menu or Cmd-Q on Mac.

Scroll is being reset in 6.7 when it is not in 6.5.2

Text wraps when field width set to formattedWidth

LiveCode application takes up 98% of processor

Printing a field with listbehaviour set to true makes gray background

Specific bug fixes (6.7.0–rc-3)

OS X standalone can’t be run out of app bundle

When Voice Over is turned on

iOS 8 ask and answer dialogs do not handle rotation correctly

Crash when opening stack

wait loop not being broken

mobilepickcontact works under ios 7 but not under ios 8

Screenshots taken in landscape view are rotated by 90 degrees on iOS 8

mobileFindContact fails silently on iOS 7.1 and higher

Simulator launches with incorrect version

Copy files do not work with the iOS 8 simulator

Changes to [[Player]] in preOpenCard are visible to user

Extra undo of paint tools crashes live code

keydown event not getting passed to mainstack in modal dialogs

[[Player]] Shift + click in controller sets showSelection to true

menuPick not triggered under certain conditions

Threaded rendering crash

Stack location reported incorrectly if mouse released while dragging window

Fix OSX specialFolderPath("asup")

Pull down menus do not work properly in modal dialogs

If an error is encountered after a drag-and-drop

4 inch iPhone apps do not use the full screen

PDF printing does not work correctly on iOS 8.

Referenced image fails to load in 6.7.0 RC2 and 7.0 RC1

Manifest file not needed in standalone bundle

mobilePick and mobilePickDate do not work in iOS 8

revPrintField clips document under some circumstances

RGB imageData values (charToNum) are different on Mac / Windows

Go stack in window displays new stack before before preopenStack/preopenCard messages are triggered

Mark the installer as retina-capable

mobilePickPhoto camera view is rotated on iPad when in landscape or in portraitUpsideDown

Image file color profiles don’t seem to be handled correctly

Fix post-install launching on Linux

The effective screenrect returns incorrect values when hiding/showing keyboard on iOS
Specific bug fixes (6.7.0–rc–2)

13354 Memory leak when setting the text of an image
13455 Non-executable file redirection on Mac
13447 Project Browser control layer display
13445 Crash due to multicore rendering.
13442 rawKeyDown for arrowKey up and down fires twice when holding altKey down
13429 Fix crash when using magnifier tool.
13423 Dead char followed by an invalid character sends an invalid RawKeyDown message
13417 IDE systemVersion comparison no longer works with Yosemite
13398 Sample - Book Library.runCode edit and delete features broken
13393 playLoudness should be rounded to nearest integer rather than down
13389 [[Player]] Can't make left to right selection
13374 LiveCode hangs when setting the text of a field with HGRID and a hidden last line
13372 Control-Tab key combos not being passed to rawKeyDown
13362 Script editor opens revmenubar script when no other stack is open
13354 App store no longer allows use of deprecated
13343 Cannot install Android standalone on some devices
13338 [[Player]] - Shift + Space Bar whilst focus is on player should create selection
13328 Setting menuhistory with a stack panel sends invalid menuPick second parameter
13310 [[Player]] add new property loadedTime to the dictionary
13309 [[Player]] setting playLoudness by script does not update speaker icon
13306 [[Player]] loadedTime initially has the value of last movie
13295 Image distortion when setting imageData to the imageData.
13286 [[ CEF browser ]] Closing a stack without closing the browser
13278 Fix a button focus issue
13222 Moving graphic while editing gradient causes artifacts
13199 [[Player]] new controller does not have playRate scrollbar control
13092 [[ Player ]] Status property requested for player object
13048 Touch on Movie does not work
13006 [[Player]] Make controller scrollbar slightly wider
12830 [[Player]] Movie not updated instantly when using QT player and alwaysBuffer is true
10703 import snapshot from object is broken

Specific bug fixes (6.7.0–rc–1)

13284 Mouse is still inside the window even when resizing
13279 rawKeyDown passes the wrong keycode if Ctrl is pressed.
13273 templateImage framecount is not accurate for animated GIF
13272 Setting the scrollbars property of a CEF browser to false has no effect.
13270 SSL doesn't work with MySQL driver on Android and iOS.
13268 [[Player]] empty player shows image of last video when loading new video
Thumb does not update properly when movie is playing in some circumstances.

App crashes when showing referenced images on Android devices

Visual effect push problem

htmlText with many nested styles can cause a crash.

Crash when rendering Mac themed scrollbars

[[Player]] Player object retains callbacks even though callbacks set to empty

Test System crashes reliably

Polygon markers draw incorrectly

Artifacts can appear in bitmap effects when multicore rendering is used.

Polyline with same starting point as ending point draws as degenerate dot in PDF printing.

Can't type in output field of message box

textFont in Text Formatting of inspector cannot scroll by a mouse

Hirigana input source causes LiveCode to hang when entering 'h' then 'a'.

[[ Player ]] LC hangs when you open a stack with a player with filename that does not exist

FIX: flip graphic horizontally and vertically for complex graphics

iOS standalone building fails with "cannot find valid identity"

[[ Player ]] Printing players doesn't work

Text is clipped when printing to PDF from OS X

Palettes not observing decorations under certain circumstances

Setting currentTime of a player in response to a currentTimeChanged message can cause a hang.

[[Player]] progress of movie downloaded/playable not indicated in controller well

Add Hi-DPI support option to Windows standalone settings dialog.

File->Exit should be File->Quit

Toggle usePixelScale property

flip graphic gives erroneous results with complex graphics

Specific bug fixes (6.7.0–dp–9)

Auto detecting WPAD on OS X added

WPAD URLs attempted to be used as proxies on OS X

Auto config pac processing broken on OS X

Crash when stack loses focus.

Tab key doesn't insert tabs into fields that should accept them.

showAll and LetterBox fullscreenmodes break on iOS

Setting the playloudness of the templateAudioClip adjusts the system volume

playloudness does not update when the user presses the mute button on keyboard

[[player]] Step back button and Callback Problems

Crash setting iconGravity - only one line needed

Crash when attempting to decode malformed arrayEncoded value.

Crash when resizing fields containing tabbed text

Quitting from the dock when app is in background waits until app is foreground to actually terminate.

centerRect property has no effect when image resizeQuality is "best"

[[Player]] cmd key + click on step forward and step backward button for rewind

crash dragging player to an empty stack
[Player] Playback is locked when alwaysBuffer is true and video is playing
[[player]] callback feature doesn't work when user moves scrubber
[[Player]] playSelection not working correctly
[[Player]] selection by shift + click Play button delayed start and stop.
Erratic behavior whilst moving a window along with excessive WindowServer CPU usage.
revVideoGrabber - revVideoGrabDialog "audio" does not bring up dialog
Sound recording returns error message
Gradient colours display incorrectly on android
libURL doesn't report actual redirect error
No error indicated if malformed URL used in url expression

Specific bug fixes (6.7.0–dp–8)
put the engine folder crashes in OSX 10.6
multi-line button label shows incorrectly - backslash n not working
Crash while drawing Mac themed controls
Cmd+Ctrl+ doesn't generate keyDown messages and causes a system beep.
The extents don't work if there are negative indicies.
Crash while resampling images
[[Player]] CurrentTimeChanged message not sent by 'step forward' or 'step backward' buttons
[[Player]] updating of playRate by controller buttons inconsistent
Crash while resizing field
[[Player]] making selection with shift + click player thumb inconsistent results
menupick message not sent from popup stacks
Wrong mouseLeave messages if the stack is pixelScaled or scaled in any fullscreen mode.
uuid: not enough randomness available
high dpi images don't fill image rect if scaled images don't have the correct size
Setting the label of a button after the unicodeLabel can cause an infinite loop in some cases.
Accented characters in string passed to revSpeak cause nothing to be spoken.

Specific bug fixes (6.7.0–dp–7)
Player doesn't wait to make sure currentTime has updated to where it has been set to.
put the engine folder behaves inconsistently
Linux GTK spinboxes were inverted
currentTimeChanged not sent when dragging controller thumb.
If an object is in a group then mouseStillDown will not fire if it is in a behavior of the object.
ImageData copying to offscreen images is broken
Native encoded characters > 127 don't appear in menuPick string.
Modifying selection by shift + click in the controller doesn't work correctly.
Controller not updated when setting currentTime by script.
selectionChanged message not sent when player selection changes via controller.
Empty selection results in selection of full movie.
Crash when changing focus inside openField handler.
Image colours not showing correctly on Android device
Changing windowShape leaves artifacts behind.
Popup menus always highlight first item on Mac
Player incorrectly reports timeScale and duration when using AVFoundation.
Setting the rect of a player object doesn't work correctly if the stack is pixelScaled or scaled in any fullscreen mode.
Development -> Plugins opens the wrong plugin
ctrl z and edit -> undo are not working
Player messages aren't sent correctly.
Player won't play video in certain circumstances.
Print dialogs do not keep new settings if displayed as sheet.
play stop command does not work.
Text does not align at correct tabStops in some cases.
Setting the text style in the Text menu does not update correctly
LiveCode Server process does not end when run as CGI
The contents of a folder added to the Copy Files section of the Standalone Application Settings are not copied during a standalone build when it's a subfolder of the defaultFolder and the defaultFolder contains the main stack file.
Attempted to arrayDecode non-arrayEncoded data can cause a crash.
LiveCode server fails to read from https URLs on Fedora
revAvailableHandlers works on password protected stacks.
Clipping incorrect when printing to PDF.
NUL bytes in fields will cause printing to pdf to fail silently.
Setting the defaultNetworkInterface more than once causes instability.
Autocomplete text in Message Box does not line up with text you are typing
File view in file dialog doesn't update if you change the filtered types.
Use 64-bit file APIs on Linux
scriptLimits no longer enforced
Clear previous highlight before drawing Linux GTK button highlight

Specific bug fixes (6.7.0–dp–6)
Closing a modal dialog causes a pause until a system event occurs.
Stacks can receive mouseDown before resumeStack.
Engine crashes when moving stack after closing its parent
IDE: Development > Rev Online != menubar > user samples
iOS standalones won't build due to missing template files.
Showing a modal dialog confuses mouse state.
Popup Menu crashes LiveCode 6.7.0 DP5 standalones
centerRect and iconGravity need to be added to the properties
Certain images have red & blue swapped when deployed to iOS
the enabledTracks of a player is formatted incorrectly.
Wrong button sent to mouseDown / mouseUp when targeting a card.
Inspector opening at odd location
Shift-click on play button doesn’t create selection.
Clicking in the well of a player with a selection should clear the selection.
Make sure controller is redrawn after setting selection by script.
Behavior of shift-click in well on player not correct.
RevVideo Grabber preview in wrong location after resizing stack
Android stack drawn at wrong scale until forced redraw of object
copying imageData between images can lead to unpredictable results
6.7.0-dp-5 Mac installer fails if system security permissions are set
playStopped sent for player with no filename
setting the effective rect of a stack with vscroll > 0 doesn’t work properly
Focus gets confused if focus changes in response to a suspendStack message on Mac
iOS font selection does not work correctly.
Browser: Closing revBrowserCef when downloading makes LiveCode unstable
Tool Palette location forgotten between launches
pixelScale global property not documented
In Dictionary the search box is only apparently selected when you come from Code Editor

Specific bug fixes (6.7.0–dp-5)

Player: scaling a player down causes controller to get confused
Player: hilite handles do not use fill length of the bar
arrayDecode crashes on linux with certain input.
answer file with type doesn’t work correctly if only one type is specified.
Windows position in the wrong place when constrained by the windowBoundingRect.
Player callbacks not operating correctly.
[[Player]] Selection thumbs should not be visible when selection duration is 0
[[Player]] Selection not created when clicking shift and dragging player thumb
[[Player]] SelectionStart and SelectionFinish handles too large and can obscure player thumb
[[Player]] Selection indicator does not align with the selection thumbs
[[Player]] loop goes to beginning of movie not selection start time when playSelection is true
[[Player]] controller icons not updated when keyboard shortcuts used to control playback
[[Player]] put the tracks of player 1 does not work properly
message box is slow
Go stack in window [windowId] doesn’t work.
On Mac
The centerRect image property doesn’t handle hi-res images correctly
Plugins don’t load in revBrowserCEF on OSX
IDE stacks white rather than grey
After dragging onto a field when LiveCode is in the background
setting dragData[files] does not work
The effective rect of a stack with vscroll > 0 is incorrect
Player: Setting player size to < 132 width breaks some controller elements
Player: dragging the in selection hilite marker moves out marker also
Player: setting the filename to a local file that doesn’t exist crashes LC
LiveCode 6.7.7-rc-4 Release Notes 9/11/15

Player: Setting the "in" marker for selection playback beyond the start point of player sets marker to unexpected value

Player: setting the filename to a URL that isn't a video crashes LC

[[Player]] Selecting "track" from a players property inspector

Player: Can't select any audio files

Player: Click outside of a selection allows video to be played outside selection

QT-related features don't work.

Player: Progress circle and end hilite don't light up

Shortcuts: the uncomment script shortcut cmd _ does not work

Player: First frame of video is not loaded immediately when filename is set

Player: File chooser doesn't filter all available video formats

Player: Can't drag out or create a player in script

Player: Hiding and showing resized player changes player size to original size

Unable to use edit mode when video is playing with new player object

zipalign tool not found during standalone build after update to Android SDK tools

Incomplete stack drawing when opening stack with acceleratedRendering on retina display

Project Explorer not updating after stack was changed to substack

Submenus of popups don't send menuPick on selection.

Fix sending of mouseRelease messages with new platform layer

Editing image then switching card and saving causes stack corruption

CEF browser crashes if htmltext is set to empty

Images don't appear or are clipped when printed to PDF

Some fonts have accents cut off on capital letters on Mac.

Blocking socket calls always timeout.

Can't deselect MacOS build

File and folder dialogs incorrectly use the topStack to sheet against

CEF browser pauses frequently when there is no other activity on the stack

Extra mouseMove with incorrect co-ordinates sent after mouseEnter when changing windows.

WebAuthenticationPanel class in OSX revbrowser conflicts with same class in WebKit library

File handle leak on Mac

Shell command does not accept spaces despite being quoted (Windows)

Multiple moves created whilst lock moves in effect fail to be synchronized.

Crash when fetching the alphadata of a resized image

Entries in the Project Browser won't expand

Cursor does not change correctly when over a revBrowser[CEF] instance.

minHeight setting on Mac includes title bar height when it shouldn't.

CEF browser returns incorrect values for rect property

Instability when using revAppendXML and revCopy/MoveRemoteXMLNode.

Use sub-pixel positioning for laying out text within fields

revBrowser placed incorrectly when dpi scaling enabled on Windows

Redraw slowdown in 6.7 (regression)

setting effective rect to working screenrect fails

Screen updates occur during 'menu update' mouseDown message causing pauses when updating menus on first click.

Pasting text into a field can sometimes cause strange selection behavior.

'listIndent' attribute does not round-trip through htmlText
12567 Connecting to an HTTPS URL via a proxy fails if libURLSetVerification is true
12566 Tunnelled proxies do not authenticate correctly
12557 Objects which are adjacent don't necessary appear so at non integral scale factors.
12556 The rtfText does not represent 'metadata' tags correctly.
12549 Hiding / deleting a stack doesn't update the mouseStack when it should.
12543 Standalone engine still links to QTKit / QuickTime.
12529 LC 6.7 dp4 plays an imported wav only once
12528 Project Browser does not scroll down to show everything
12524 Hiding player controller stretches movie image vertically
12523 [[Player]] Setting playRate of player has no effect on playRate
12512 player currentTimeChanged message does not include time parameter
12506 Instability with manipulating QTKit players.
12501 Setting callbacks in player causes crash
12495 [[ Bug 12495 ]] Animating windowShape does not work properly on Mac.
12481 Various actions on players (such as hiding and showing) prevent it from working properly.
12479 Maximum number of paragraphs which can be set with styledText is 64k
12470 Terminal window appears when accessing User Samples on Windows
12468 Middle button paste doesn't work correctly in other apps when LiveCode has the selection on Linux.
12467 Changing decorations causes no cursor to appear over a stack.
12463 Queuing too many pending messages causes slowdown and random crashes.
12462 Maximize button in Windows title bar doesn't use full screen under high-dpi Windows
12458 Crash when reading invalid image data
12451 Popup windows and combo-box menus don't disappear when they should.
12437 Cursor changes incorrectly for top pixel of borderless windows on Mac.
12436 import snapshot does not always display crosshair
12404 When using import/export snapshot from screen in non-interactive mode
12401 Browser: revBrowserSnapshot issues in 6.7.0 DP3
12388 Drag-drop does not work if 'private' data type is used on Mac.
12370 Key codes are mapped differently resulting keyboard shortcuts not acting correctly with non-English keyboard layouts.
12363 Cmd-Z shortcut missing from "Undo" menu item
12354 AcceleratedRendering causes double-sized stack controls on Retina displays
12351 Crash on write then read until EOF on driver
12339 mouseRelease message sent after selection from popup menu.
12321 On Windows 7 Fullscreen set to false does not return to previous size
12312 VideoGrabber doesn't work on Mac.
12303 Setting the text of a field chunk should not clear the paragraph styles of an empty line.
12297 Windows opened in popup mode have decorations.
12225 Menubar in application makse revBrowser misaligned
12156 Single line message box field is too large
12055 mobileVibrate not vibrating when passed a variable
12014 Stack with iOS deployment checked(mac) when opened on windows system does not allow you to disable this deployment option. This in turn breaks Android APK generation
11928 Inconsistencies in behavior when doing 'delete the selectedChunk'.
11878 Pasting with the middle mouse button on linux doesn't work correctly.
Menus in Script Editor have testSize too small

Ensure that spaceAbove area is redrawn when hilite changes.

Dictation is unstable on Mac.

Buttons in Ask

Help menu: duplicate name

Command click in project browser

Duplicate Menu and Replicate dialog lock messages

Project Browser disappears in dual screen

When setting the styledText of a range

ask dialog icon is a button with its autohilite set to true

Specific bug fixes (6.7.0–dp–4)

revBrowserOpenCEF doesn't send browserDocumentComplete when setting htmltext

Standard menu items with tags (Cut/Copy/Paste etc.) do not enabled/disable correctly.

Crash when calling revBrowserClose on revCEFBrowser

Specific bug fixes (6.7.0–dp–3)

Scrollbars don't get keyboard input when focused if no field is on the card on Mac.

Menu items are all disabled in modal dialogs on Mac.

Make sure invisible stacks still have a valid windowId on Mac.

New CEF-based revBrowser has no authentication dialog

Pasting LiveCode controls from other copies of LiveCode doesn't work

formattedWidth/Height of player objects is incorrect

Command key shortcuts don't work if revBrowser has focus on Mac.

Stack location is always constrained to screen on Mac.

Resize stack cursor not set sometimes.

resizable and maximize decorations not honoured correctly on Mac.

Answer dialogs do not respond to mouse input if opened from a menu on Mac.

dragDrop sometimes doesn't work when dropping from other applications (on Mac).

dragImageOffset is incorrect horizontally on Mac.

Crash after releasing mouse on stack after drag-drop operation on Mac.

File > Exit menu item not hidden on Mac.

Player object does not respond properly to edit tool.

Accelerated rendering mode doesn't work correctly on Mac if using coregraphics mode.

revBrowser doesn't grab mouse focus on Mac.

the qtVersion returns empty

Beach-ball shows in tight loops on Mac.

Resize cursor does not always display in window edges on Mac.

NullPointerException on device logcat when app was launched and closed if in-app purchasing box is not ticked

Import / export snapshot from rect of window ignores rect on OSX

"Export snapshot ... from window ..." wrongly scales snapshot rect

Setting a player filename to a url causes hang.

moveStack messages are not sent continously on Mac.

rawKeyDown sends incorrect code for shifted keys on Mac.
"export snapshot of window ..." locks up on OSX
Changing an applied pattern causes the IDE to crash
Setting hidePalettes to false does not work on Mac.
Make sure 'unlock cursor' takes immediate effect
No default menubar in standalones on Mac.
Clicking on a cell of a table field causes the IDE to crash
Arrow keys don't work in CEF browser on Mac.
Players do not render at correct size on Retina Macs.
Non-CEF browser not working in modal dialogs on Mac.
The mouse cursor does not hide when typing on Mac.

Specific bug fixes (6.7.0–dp–1)

Setting the label of an option or combo-box does not update the menuHistory.

pixelScaling not enabled on Windows Commercial edition

IDE changes

HTTPS through proxy (6.7.1–rc–1)

The desktop version of LibURL has been updated to support fetching HTTPS URLs through a proxy server.

Proxy automatic configuration support (6.7.1–rc–1)

LibURL has been updated to add support for proxy auto configuration files (PAC). If no global proxy server has been set using the HTTPProxy property, when fetching a URL, LibURL will attempt to parse the systems .pac file in order to extract the proxy server (if any) to use for the given URL.

Specific bug fixes (6.7.7–rc–3)

iOS Version resets to 5.1.1

Specific bug fixes (6.7.7–rc–1)

Mobile Support JDK path not updated after changing it
Setting spaceAbove or spaceBelow on the first line of a table breaks it
Magnify palette cannot be used in a standalone
cell editing field in basic table field
Find and replace does not support arrow keys
Project Browser forces the unlocking of the cursor
message box does not auto-scroll
Editing a table field cell shows focus at the wrong place when margins are changed.

Specific bug fixes (6.7.6–rc–1)

mobile datagrid coding interferes with mobile scrollers
Certain update notification settings not saved
Needs to deactivate twice the messages
Save As replaces spaces with underscores in file name
15240 datagrid mobile does not handle multiple selected lines
14802 ProjectBrowser Disclosure Triangles Not Responding
14788 Deleting lots of selected controls redraws after each delete
14475 Project Browser Can't Hold Size
13953 book database demonstration doesn't work on Linux
          Context menu item "Go to Definition..." always disabled in IDE scripts and names of handlers
          which are defined in libraries
10722
Specific bug fixes (6.7.5–rc–3)
15329 Make sure loading/unloading revNoMessagesLibrary results in insertion into/removal from the
          frontscripts
15298 Cannot select minimum version iOS 8.3
Specific bug fixes (6.7.5–rc–1)
15135 [IDE] Remove dialog box asking for the version of Xcode
15131 Local notifications check box is not refreshing properly
15054 revShortCutsLibrary not managing its position in the message path correctly
Specific bug fixes (6.7.4–rc–1)
14400 App requests notification permission when it isn't configured to use them
Specific bug fixes (6.7.2–rc–1)
14341 Menu Builder can't create menu items with both Shortcut and Tag
Specific bug fixes (6.7.1)
13947 Wrong image in "About Livecode" for LC 6.7 Community
Specific bug fixes (6.7.1–rc–4)
14155 Menu Builder does not work reliably: tags disappear
Specific bug fixes (6.7.1–rc–3)
14072 'About' menu on the Mac omits further menu item text
Specific bug fixes (6.7.1–rc–1)
13190 iOS standalone building fails with "cannot find valid identity"
13172 Auto detecting WPAD on OS X added
13171 WPAD URLs attempted to be used as proxies on OS X
13170 Auto config pac processing broken on OS X
13118 Add Hi-DPI support option to Windows standalone settings dialog.
13117 multi-line button label shows incorrectly - backslash n not working
12978 Development -> Plugins opens the wrong plugin
12968 ctrl z and edit -> undo are not working
12928 IDE: Development > Rev Online != menubar > user samples
12902 iOS standalones won't build due to missing template files.
12879 Inspector opening at odd location
The contents of a folder added to the Copy Files section of the Standalone Application Settings are not copied during a standalone build when it's a subfolder of the defaultFolder and the defaultFolder contains the main stack file.

message box is slow
libURL doesn't report actual redirect error
Player: Can't select any audio files
Player: File chooser doesn't filter all available video formats
zipalign tool not found during standalone build after update to Android SDK tools
Project Explorer not updating after stack was changed to substack
Can't deselect MacOS build
Entries in the Project Browser won't expand
Redraw slowdown in 6.7 (regression)
Tunnelled proxies do not authenticate correctly
Project Browser does not scroll down to show everything
Terminal window appears when accessing User Samples on Windows
Cmd-Z shortcut missing from "Undo" menu item
Autocomplete text in Message Box does not line up with text you are typing
Buttons of menu type can't be inspected if first created object
Single line message box field is too large
Stack with iOS deployment checked(mac) when opened on windows system does not allow you to disable this deployment option. This in turn breaks Android APK generation
Menus in Script Editor have testSize too small
Hi-res iPad icon size incorrect
Android app name can't contain unicode text
pixelScale global property not documented
LiveCode cannot resolve path to Android Runtime folder when building Android standalone
Default font in standalones causing PDF to render badly
Script editor line numbers are not displayed
Buttons in Ask
Linux "Open Stack..." dialog does not show "All Files" option.
Internal Livecode image with ID 1105 creates ID problem
Can't quit LiveCode after setting the securityPermissions to "network"
Help menu: duplicate name
In Dictionary the search box is only apparently selected when you come from Code Editor
Command click in project browser
Answer folder for XCode app bundle has a typo
messagebox syntax for invoking functions is broken
Project browser does not expand cards of substacks
menu/Text: size does not work
Start centre missing from IDE
iOS 7 icon entries missing from bundle plist
Duplicate Menu and Replicate dialog lock messages
iOS 7 Hi-Res iPad icon size requirements incorrect.
Unable to set iPad status bar visibility independently of iPhone status bar visibility.
The width of the fields in the message box is wrong.
Basic table field with multiple tabStops stops inspector from working.
Preferences show wrong panel
Populating a datagrid from a card
Including iOS externals in Android projects (and vice versa) causes standalones to fail.
DataGrid throws error when setting show vscrollbar to false in preOpenControl on mobile.
Moving a Layer between two groups in Project Browser
No error message when external not found when deploying to simulator
Project Browser denies existence of existing controls!
   Menu Builder Disable Item problem. Unicode characters were not being handled correctly when disabling/enabling menu items.
Plugin settings refers to Revolution. This has been updated to refer to LiveCode.
iOS minimum version in standalone builder is 3.1.3 (should be 4.3).
Script Editor doesn't include hash symbol in token delimiters
mobileComposeMail attachment missing in Android
LiveCode IDE breaks when opening a particular PDF file - uncaught exception fetching the properties of a DataGrid
Find Malfunction in Script Editor for both >*/ and double-quote ("")
The Project Browser does not update when a group is ungrouped.
Autoscript "compile error" with new menu items
User Guide entry under Help menu is missing
The PDF Printing external is not being included in Windows Standalones.
app identifier field in standalone applications settings on iOS and Android allows return characters.
Project Browser disappears in dual screen
Selecting regular polygon graphic fails to update title of Property Inspector
Mobile stack saved as iOS retains iOS settings on Windows. As the option is disabled and cannot be unchecked on Windows the standalone builder will disregard this choice on Windows. Selecting another platform to build for will still automatically unselect iOS.
Menu Text > Font > Size > Other... does nothing. Now asks the user to input a text size.
HTTP "get URL" omits port number from HOST header
revUniqueID - the revUniqueID of the child controls of a group do not get updated when the group is cloned or duplicated.
Save this stack as syntax broken in the message box
iPad retina landscape image size warning incorrect.
Indenting of scripts can go wrong
Thumb and scrollbar appear over top of DataGrid
setting a cprop with quotes loses data
ask dialog icon is a button with its autohilite set to true
LiveCode Builder changes
Previous Release Notes

7.0.6 Release Notes  http://downloads.livecode.com/livecode/7_0_6/LiveCodeNotes-7_0_6.pdf
7.0.4 Release Notes  http://downloads.livecode.com/livecode/7_0_4/LiveCodeNotes-7_0_4.pdf
7.0.3 Release Notes  http://downloads.livecode.com/livecode/7_0_3/LiveCodeNotes-7_0_3.pdf
7.0.1 Release Notes  http://downloads.livecode.com/livecode/7_0_1/LiveCodeNotes-7_0_1.pdf
7.0.0 Release Notes  http://downloads.livecode.com/livecode/7_0_0/LiveCodeNotes-7_0_0.pdf
6.7.2 Release Notes  http://downloads.livecode.com/livecode/6_7_2/LiveCodeNotes-6_7_2.pdf
6.7.1 Release Notes  http://downloads.livecode.com/livecode/6_7_1/LiveCodeNotes-6_7_1.pdf
6.7.0 Release Notes  http://downloads.livecode.com/livecode/6_7_0/LiveCodeNotes-6_7_0.pdf
6.6.0 Release Notes  http://downloads.livecode.com/livecode/6_6_0/LiveCodeNotes-6_6_0.pdf
6.5.2 Release Notes  http://downloads.livecode.com/livecode/6_5_2/LiveCodeNotes-6_5_2.pdf
6.5.1 Release Notes  http://downloads.livecode.com/livecode/6_5_1/LiveCodeNotes-6_5_1.pdf
6.5.0 Release Notes  http://downloads.livecode.com/livecode/6_5_0/LiveCodeNotes-6_5_0.pdf
6.1.2 Release Notes  http://downloads.livecode.com/livecode/6_1_2/LiveCodeNotes-6_1_2.pdf
6.1.1 Release Notes  http://downloads.livecode.com/livecode/6_1_1/LiveCodeNotes-6_1_1.pdf
6.1.0 Release Notes  http://downloads.livecode.com/livecode/6_1_0/LiveCodeNotes-6_1_0.pdf