LiveCode 7.1.3-rc-1 Release Notes

Table of contents

Overview
Known issues
Platform support
  Windows
  Linux
  Mac
  iOS
  Android
Setup
  Installation
  Uninstallation
Reporting installer issues
Activation
Multi-user and network install support (4.5.3)
Command-line installation
Command-line uninstallation
Command-line activation
Engine changes
  Installer no longer required on OSX
  DataGrid added to the Standalone Settings script libraries list
  Ensure 'the effective rect of stack' is more accurate on Linux
  Application Transport Security (ATS)
  Android Hardware Acceleration
  *launchDataChanged* message on Android
  Add beta testing entitlement to iOS apps standalone settings
  Ability to set the dontUseQT property for a player object (Windows and OSX)
  background audio on iOS
  New "mirrored" property for the OSX player. Applies to both QTKit and AVFoundation player.
  New 'readyForDisplay' ios player property
  Windows default videograbber is now DirectX
Xcode 7.0
  Some arrays encoded in 6.7 format from 7.0 won't load into 6.7.
  revCapture - revCaptureListVideoCodecs() results in crash
  mobileLaunchData function
    Available information
    the *commandName* and the *commandArguments*
Multimedia on MacOS with AVFoundation
Mobile Sockets
  Objects are only deleted on idle
Specific bug fixes (7.1.3-rc-1)
Specific bug fixes (7.1.2-rc-3)
Specific bug fixes (7.1.2-rc-2)
Specific bug fixes (7.1.2-rc-1)
Specific bug fixes (7.1.1)
Specific bug fixes (7.1.1-rc-4)
Specific bug fixes (7.1.1-rc-3)
Specific bug fixes (7.1.1-rc-2)
Specific bug fixes (7.1.1-rc-1)
Specific bug fixes (7.1.0)
Specific bug fixes (7.1.0-rc-3)
Specific bug fixes (7.1.0-rc-2)
Specific bug fixes (7.1.0-rc-1)
Specific bug fixes (7.1.0-dp-1_release)
Specific bug fixes (7.1.0-dp-1)

IDE changes
Specific bug fixes (7.1.2-rc-1)
Specific bug fixes (7.1.1-rc-1)
Specific bug fixes (7.1.0-rc-2)
Specific bug fixes (7.1.0-dp-1)

LiveCode Builder changes

Previous Release Notes
Overview

The LiveCode engine has undergone a large quantity of changes for the 7.0 release. The way values of variables are stored internally has been changed - in particular where before the engine used C-strings, it now uses a reference counted MCStringRef type. Every bit of code that displays text in LiveCode has been updated, and all the platform-specific API functions that manipulate characters now use the Unicode versions; as a result LiveCode is now fully Unicode compatible.

The implementation of Unicode compatibility necessitated a change to the stack file format, which means stacks saved in 7.0 format are not compatible with earlier versions of LiveCode. However you can still save stacks in legacy formats using the dropdown menu in the Save As... dialog.

The other significant change to engine internals is the work done on syntax refactoring. The code that deals with statement execution, function evaluation and property access has been cleaned up and separated out from the parsing code, and moved into distinct modules based on functionality. This represents a major first step towards being able to implement Open Language.

Known issues

Every effort has been made to ensure that externally, the engine behaviour is identical to the current unrefactored release. In other words, users should not notice any difference in functionality in their existing stacks.

- The installer will currently fail if you run it from a network share on Windows. Please copy the installer to a local disk before launching on this platform.
- The engine files are much larger than previous versions due to inclusion of ICU data

Platform support

The engine supports a variety of operating systems and versions. This section describes the platforms that we ensure the engine runs on without issue (although in some cases with reduced functionality).

Windows

LiveCode supports the following versions of Windows:

- Windows XP SP2 and above
- Windows Server 2003
- Windows Vista SP1 and above (both 32-bit and 64-bit)
- Windows 7 (both 32-bit and 64-bit)
- Windows Server 2008
- Windows 8.x (Desktop)

Note: On 64-bit Windows installations, LiveCode runs as a 32-bit application through the WoW layer.

Linux

LiveCode supports Linux installations which meet the following requirements:

- Supported CPU architectures:
- 32-bit or 64-bit Intel/AMD or compatible processor
- 32-bit ARMv6 with hardware floating-point (e.g. RaspberryPi)

- Common requirements for GUI functionality:
  - GTK/GDK/Glib 2.24 or later
  - Pango with Xft support
  - esd (optional, needed for audio output)
  - mplayer (optional, needed for media player functionality)
  - Icms (optional, required for color profile support in images)
  - gksu (optional, required for privilege elevation support)

- Requirements for 32-bit Intel/AMD:
  - glibc 2.3.6 or later

- Requirements for 64-bit Intel/AMD:
  - glibc 2.15 or later

- Requirements for ARMv6:
  - glibc 2.7 or later

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

**Note:** The requirements for GUI functionality are also required by Firefox and Chrome, so if your Linux distribution runs one of those, it will run LiveCode.

**Note:** It may be possible to compile and run LiveCode Community for Linux on other architectures but this is not officially supported.

### Mac

The Mac engine supports:

- 10.6.x (Snow Leopard) on Intel
- 10.7.x (Lion) on Intel
- 10.8.x (Mountain Lion) on Intel
- 10.9.x (Mavericks) on Intel
- 10.10.x (Yosemite) on Intel
- 10.11.x (El Capitan) on Intel

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

### iOS

iOS deployment is possible when running LiveCode IDE on a Mac, and provided Xcode is installed and has been set in LiveCode **Preferences** (in the **Mobile Support** pane).

Currently, the supported versions of Xcode are:

- Xcode 4.6 on MacOS X 10.7
- Xcode 5.1 on MacOS X 10.8
- Xcode 6.2 on MacOS X 10.9
- Xcode 6.2, 6.4 and 7.2 on Mac OS X 10.10
- Xcode 7.2 on MacOS X 10.11

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

We currently support the following iOS Simulators:

- 5.1
- 6.1
- 7.1
- 8.2
- 8.4
- 9.2

Android

LiveCode allows you to save your stack as an Android application, and also to deploy it on an Android device or simulator from the IDE. Android deployment is possible from Windows, Linux and Mac OSX.

To enable the deployment to an Android device, you need to download the Android SDK and to install the following component, using the Android SDK Manager:

- Android SDK Platform for Android 4.0.3 (API 15)
- Android SDK Platform Tools
- Android SDK Build Tools

You also need to have Java JDK 1.6 installed on your machine (or openjdk-6 on Linux).

Once you have set the path of your Android SDK in the LiveCode Preferences > Mobile Support, you can deploy your stack to Android devices running Android >= 2.3.3.

Android Watch is not officially supported at the moment.
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 7.1.3-rc-1`
- Linux: `/opt/runrev/livecode-7.1.3-rc-1`
- Mac: `/Applications/LiveCode 7.1.3-rc-1.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 7.1.3-rc-1`
- Linux: `~/.runrev/components/livecode-7.1.3-rc-1`
- Mac: `~/Applications/LiveCode 7.1.3-rc-1.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 LiveCode install. e.g.

```
cd /opt/runrev/livecode-7.1.3-rc-1
```

- execute the `setup.x86` (or `setup.x86_64` for 64-bit LiveCode) 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@livecode.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/
  - **Windows Vista/7:** `<users folder>/<user>/AppData/Local/RunRev/Logs`
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 Mac, Linux and Windows. When invoked in this fashion, no GUI will be displayed, configuration being supplied by arguments passed to the installer.

On the three platforms, the command is of the following form:
<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 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 uninstallation

It is possible to uninstall LiveCode from the command-line on Windows and Linux. When invoked in this fashion, no GUI will be displayed.
On both platforms, the command is of the following form:
<exe> uninstall noui
Where <exe> is .setup.exe on Windows, and .setup.x86 on Linux. This executable, for both of the platforms, is located in the folder where LiveCode is installed.
In the same way as for the installer, there is no authentication possible with the command-line variant of the uninstaller. It means that you need to run the uninstaller as administrator to uninstall LiveCode, if it was installed in an admin-only location.
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:
<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 LiveCode customer accounts area.
This action can be undone using the following command:
<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:
start /wait <livecodeexe> activate -file license -passphrase phrase
start /wait <livecodeexe> deactivate
On Mac, you need to do:
“<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.
Engine changes

Installer no longer required on OSX (7.1.3-rc-1)

The installation process on OSX has been updated to no longer require an installer. Instead, LiveCode now installs like most other OSX applications: drag the app from the DMG to your Applications folder.

Similarly, there is no longer an uninstaller; drag the app to the Trash to remove.

The IDE app bundle is now signed, which means it can now be used in certain situations that require verification using Apple's GateKeeper technology. As such, you should not modify the contents of the app bundle (for example, editing the IDE stacks) if you use the IDE for purposes that require verification as the signature will no longer be valid.

DataGrid added to the Standalone Settings script libraries list (7.1.2-rc-1)

Formerly, the DataGrid library was only included in a standalone application if the stack saved as a standalone was using a DataGrid.

This have been causing issues, in case the stack saved does not use DataGrid, but loads a stack which uses it: the DataGrid library was not saved with the standalone, and the DataGrid would not work in the loaded stack.

To tackle this issue, we have added 'DataGrid' in the list of Script Libraries in the Standalone Settings. You can now force the inclusion of the DataGrid library, to ensure that any stack loaded by the standalone can use DataGrids.

Ensure 'the effective rect of stack' is more accurate on Linux (7.1.2-rc-1)

With the move to GDK since 7.0 there is a better method for computing the effective rect of a window. The engine has been updated to use this method, rather than the heuristic which was there before.

Application Transport Security (ATS) (7.1.1-rc-3)

Apple introduced in iOS SDK 9.0 the Application Transport Security which 'enforces best practices in the secure connections between an app and its back end' (see the technical notice).

The most noticeable effect on applications created using Xcode 7.0 (and following versions) is that URLs using HTTP protocol are no longer considered valid, and the iOS engine will not process them; only URLs using HTTPS are deemed secure enough, and will be loaded.

This means that nativeBrowser cannot load a webpage such as http LiveCode, but will happily load https LiveCode. The same applies to the LiveCode function url.

To allow our users to create apps letting Web navigation accept unsecure webpages, we added a checkbox Disable ATS in the Standalone Settings for iOS, in the Requirements and Restrictions section. If you check this box, then ATS will be disabled, and the application can load Webpages using HTTP (as it used to do).

Android Hardware Acceleration (7.1.1-rc-1)
In order to have the Native Android Browser object being able to play videos, the Hardware Acceleration of the activity must be enabled.

However, activating the hardware acceleration rendering decreases the stack rendering speed considerably due to our use of a frequently changing bitmap as the stack view.

Thus, we added an option in the Standalone Settings to let users choose between a fast-rendering application (without the Hardware Acceleration) or an application whose Native Browsers can play videos.

`launchDataChanged` message on Android (7.1.1-rc-1)

Added the `launchDataChanged` message on Android to indicate when the app was resumed to handle a new request from another app.

Add beta testing entitlement to iOS apps standalone settings (7.1.1-rc-1)

From LiveCode 6.7.8 RC 1 and LiveCode 7.1.1 RC 1, a checkbox allows users to add a beta testing entitlement to their iOS applications.

Ability to set the `dontUseQT` property for a player object (Windows and OSX) (7.1.1-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.

On OSX, the default value of the `dontUseQT` (global) property is true if the OSX version is greater or equal to 10.8. This means that any player object created will use the AVFoundation API for multimedia playback. With this new feature, you can set the `dontUseQT` property of a player to false, without changing the value of the global `dontUseQt` property. In that way you can have both QuickTime and AVFoundation players playing at the same time. This can be particularly useful for supporting some media formats or codecs that are not supported by the default AVFoundation player (for example .midi files, Sorenson Video 3, H.261 codecs etc)

background audio on iOS (7.1.1-rc-1)

A new “experimental” feature has been added, which allows audio to continue playing while the app is on the background (i.e. when switching apps). We have added a new checkbox “Background Audio (experimental)” in the Standalone Application Settings for iOS, which enables this feature by modifying the plist settings in the appropriate way.

The reason for this being “experimental” is that the engine does not yet support suspend/resume explicitly - this means that you do have to check your applications work correctly on startup / exit. In particular, if your app saves state on shutdown you might need to ensure it is saved more frequently so if your app gets terminated when in the background, data is not lost.
New "mirrored" property for the OSX player. Applies to both QTKit and AVFoundation player. (7.1.1-rc-1)

The default value of this property is "false". Setting this property to "true" makes the video frames to be displayed mirrored.

Syntax: set the mirrored of player "myPlayer" to true

New 'readyForDisplay' ios player property (7.1.1-rc-1)

We added a new 'readyForDisplay' property to the ios player. This maps to the native readyForDisplay property, and is read-only. For more details see the dictionary entry of mobileControlGet function.

Windows default videograbber is now DirectX (7.1.1-rc-1)

Since QuickTime has been deprecated by Apple for a long time and is not maintained anymore, and that it is getting more and more difficult to install QT on the newer Windows versions, the default Windows videograbber has been updated to be DirectX.

Xcode 7.0 (7.1.1-rc-1)

iOS 9.0 SDK is now part of the valid SDKs that can be used for iOS standalone application deployment, from Mac OSX 10.10.4.

If you installed the new version of Xcode at another location than the previous one, we invite you to update the Xcode in Preferences > Mobile Preferences.

All the previously existing versions of iOS supported are still supported.

Some arrays encoded in 6.7 format from 7.0 won't load into 6.7. (7.1.1-rc-1)

It was possible for an array in 7.0 to have a key that contained the empty array. When encoded in 6.7 format using arrayEncode, the resulting data would not decode correctly in 6.7 - producing a truncated result.

This has been fixed - 6.7 will now successfully load such arrays when generated from 7.0.

revCapture - revCaptureListVideoCodecs() results in crash (7.1.1-rc-1)

To palliate this problem, some getters in the library revCapture must return possibly UTF-8 encoded names (such as the codecs) to allow the script writer to set them.

In the same idea, some setters can be given UTF-8 encoded strings.

Affected getters:
- revCaptureListAudioInputs
- revCaptureListVideoInputs
- revCaptureGetAudioInput
- revCaptureGetVideoInput
- revCaptureGetPreviewImage
- revCaptureListAudioCodecs
- revCaptureListVideoCodecs
- revCaptureGetAudioCodec
- revCaptureGetVideoCodec
- revCaptureGetRecordOutput

Affected setters:
- revCaptureSetAudioInput
- revCaptureSetVideoInput
- revCaptureSetPreviewImage
- revCaptureSetAudioCodec
- revCaptureSetVideoCodec
- revCaptureSetRecordOutput

mobileLaunchData function (7.1.0-rc-2)

This new function is available on Android and returns an array containing information from the Intent object used to launch the currently running app.

Available information

- **action** - The general action the app was launched to perform.
- **data** - The data to operate on.
- **type** - The MIME type of the data provided.
- **categories** - Additional information about the action to perform.
- **extras** - An array of action-specific data set by the calling activity.

the **commandName** and the **commandArguments** (7.1.0-rc-1)

The bugs 12106 and 12018 have been longstanding issues: $0 is not the command name, but the first command argument (which does not follow the way Bash works).

To solve this issue and avoid breaking any script that uses the current way command arguments are retrieved with LiveCode ($0 being the first commandline argument instead of the command name), we introduced two functions to allow the users to access the commandline name and arguments.

- the **commandName** returns the command that has been used to start the executable
- the **commandArguments** returns a 1-based, numeric array of the commandline arguments if no index is given. Returns the arguments at this index otherwise (or empty if the index is < 1 or > number of parameters)

These functions are only implemented for desktop standalone applications and server scripts. They will return empty on mobile platforms and in the IDE.

Multimedia on MacOS with AVFoundation (7.1.0-dp-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.

**Mobile Sockets (7.1.0-dp-1)**

Socket support has been added to the mobile platforms. The following syntax has been added to the iOS and Android engines.

**Commands:**

- accept
- open socket
- close socket
- read from socket
- write to socket
- secure socket

**Functions:**

- openSockets

**Messages:**

- socketClosed
- socketError
- socketTimeout

**Properties:**

- socketTimeoutInterval

If you are secure sockets, the SSL library must be included in your standalone. To do this for iOS,
make sure the "Encryption" checkbox of "Basic Application Settings" section on the iOS screen of the Standalone Application Settings window is selected. To do this for Android, make sure the "SSL & Encryption" checkbox of "Basic Application Settings" section on the Android screen of the Standalone Application Settings window is selected.

Objects are only deleted on idle (7.1.0-dp-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 write object processing code which creates and deleted many objects.

Specific bug fixes (7.1.3-rc-1)  
(bug fixes specific to the current build are highlighted in bold, reverted bug fixes are stricken through)

17031  mobilecomposemail now returns a result code
17019  Make sure listIndex is set correctly when setting the styledText of a field
17013  Don't crash when constructing huge strings
16984  Fixed iOS Mobile Player Seek Commands being reversed
16945  Make sure mobilePick..."checkmark" behaves the same between iOS and Android
16939  Make sure large files do not EOF prematurely
16139  Don't draw leading '(' on labels for disabled tab buttons

Specific bug fixes (7.1.2-rc-3)

16932  Page Setup and Printer Dialogues now open successfully

Specific bug fixes (7.1.2-rc-2)

16889  Fix crash when going to a substack
16859  Make sure chmod is not used by Windows installer
16853  set htmtext crash if there are spaces between and "=" sign

Specific bug fixes (7.1.2-rc-1)

16772  Android: fix incorrect offset of the working screenrect
16768  Update to libpng 1.5.26
16741  The attached stack will not open in LC 7.1.1
16737  playstopped message after playing mp3 with player waits until mouse is moved
16735  Added beta entitlement for testflight
16704  android native multiline ("editable"
16700  URLs hosted on SNI servers are not supported
16661  Dict: "before" and "after" have bad examples
Filtering with wildcards broken in very specific edge case

SpecialFolderPath("resources") doesn't work in substack

Update to OpenSSL 1.0.1q

'put the commandArguments' crashes LiveCode IDE

Setting the colors of a field to 8 empty lines has no effect

Player callbacks not sent

empty is an ascii string

Checking if a variable is a color turns Unicode chars into '?'

Arrays are ASCII string

Fixed instability when using styledText with style arrays containing listStyle or textAlign entries

Platforms sections doesn't mention Android or HTML 5

mobileControlGet(..)

global variable shadowing local variable is not detected

Setting RTF text adds invalid chars in the field on Linux

setting hiliteFill of a button sets its autoArm property

Array intersect and union differences between 6.7 and 7

Linux uninstaller is named incorrectly on non-x86 platforms

Linux uninstaller is installed without executable bit set

Splitting by empty causes a hang

bitNot dictionary example incorrect

Going to a card forces a stack onscreen

Hiding a mobile player does not automatically hide the controller on Android

Incomplete RTF generated when text contains hyperlinks

Fix text layout when printing to PDF on Linux

recordCompressionTypes crash LC 7.1.1 rc 1 if dontUseQt is true

Crash on standalone shutdown

create script only stack doesn't automatically update Project Browser

the (global) linkhilitecolor is always "0"

[LC 7] mobileCreateLocalNotification with empty payload causes crash

Release Notes should clearly state supported Xcode versions

there is a folder empty returns true

Touch interaction with iOS native scrollers doesn't work until after the scroller has been scrolled.

Cannot choose to include DataGrid library in standalone settings

The controlIDs property reports empty lines after the id of an empty group.

'the folders' doesn't return ".." if there are no other folders on MacOSX

Hiliting lines in a field are not accurate or working correctly with word selection

ArrowKey left in tabbed field stuck if first column empty

Popup menu can get corrupted when click-drag out of the list

Setting the styledText wipes out listStyle

Parsing a Uint list fails in LC 7 if it contains "empty"

Strange behaviour of combo boxes after resizing

stack height keeps being changed between LC sessions

"infinity" is not always equal to "infinity"

DataGrid crashing app in sim
"subString is not in string" Description is wrong
Mac - Icons to display on ask and answer dialogs - not showing
Media player callbacks are not called with the right defaultStack
Ensure 'the effective rect of stack' is more accurate on Linux
Documentation of is a number is not consistent with the LC Engine
export snapshot not working on Android
scalefactor doesn't change player size in browse mode
listBehavior fields don't become the selectedField if focused by any means other than clicking.

Specific bug fixes (7.1.1)
LiveCode 6.7.8 (RC4) will not test launch app in iOS Simulator 9.1

Specific bug fixes (7.1.1-rc-4)
repeat for each codepoint / item / line subtlety broken
Stack location locked in y-axis when stack height greater than monitor height
Cannot deploy iOS app if name contains a '

Specific bug fixes (7.1.1-rc-3)
Unable to mobileControlSet URL with http:// URL on iOS 9.1 Simulators and Devices
First iOS standalone build with Xcode 7.0/7.1 causes "linking for arm" error
Use the server-revdb.dll external for Windows server

Specific bug fixes (7.1.1-rc-2)
Remove unneeded load-time dependence on QTKit.
When switching between windows on Mac
Successive sorts do not preserve order of equal elements.
MobileControlDelete crashes app in LC 6.7.8 (RC 1) on iOS sim 9.0
Exterals in the Copy Files of a standalone causes the app to crash in the iOS simulator

Specific bug fixes (7.1.1-rc-1)
Allow audio playing when app is on background on iOS
Can't deploy to iOS 9 Simulator when using referenced images
Save As dialog always opens in a folder deep in LiveCode's app bundle
Crash when fetching 'the shadow' property of an object.
Crash when closing button menu by clicking outside of it
revOpenDatabase missing from Dictionary
revZipOpenArchive stops working on LC 7.1 and xcode 6.4
open process / launch does not word with quotes around the executable name
Check for update does not work
Empty converted to 0 in object id chunk expression
Android audio/video controller disappears after some seconds of inactivity
Message box doesn't remember where it was placed
play stop does not work in LC 7.x on mobile
Specific bug fixes (7.1.0)
App crash during launch on iPad2 with iOS min version set to 5.1.1 (LiveCode 7.1 RC+DP)

Specific bug fixes (7.1.0-rc-3)
15898 Building 32-bit slice only causes app to crash on iPad 1

Specific bug fixes (7.1.0-rc-2)
15764 Android mobileControlCreate "player" not working
15756 various codepoint and codeunit functionality broken in standalones
15743 iOS standalone engine do not build anymore in Debug mode
15738 The "My LiveCode/Plugins" folder isn't created by the IDE when updating the BAF
15723 Incorrect wording in Business edition activation screen
15719 An error in a preOpenStack script aborted openStack
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
15684 "extras" key in array returned with mobileGetLaunchData is empty
15676 Unicode chars in a script can make the Script Editor crash on script opening

Specific bug fixes (7.1.0-rc-1)
15700 iOS sockets only accept a single connection
15692 [Docs] Dictionary entry for "flip" command is missing text
15675 IDE crashes on startup if DataTree is installed
15654 LiveCode 6.7.7 RC 1 / 7.1.0 DP 1 won't install on 10.6.8
15648 Exporting text to RTF text generates invalid output if there is the backgroundcolor has been changed
15646 Memory leak when using 'put into`
15642 revZip fails to open archive in 64-bit iOS
15633 ceiling (*ceil()*) synonym is missing from the dictionary
15625 "record sound" is not replacing existing sound file
15623 machine() returns "unknown" in LiveCode 7.x on OSX
15619 Opening sqlite on iOS return "revdberror
15602 Recursionlimit property cannot be set by script > 65535
15597 LiveCode 7 fail to export valid RTF text if lists are used
15592 memory leak in shell() on Mac
15590 Crop image within a repeat loop causes crash to desktop after a few iterations
15514 Blocking socket syntax now works in LiveCode Server.
15379 Regular Expression with binary input fails in LC 7
15200 Default cursor doesn't reset when set to empty
15096 Escape key does not dismiss the 'answer color' dialog
13923 formattedHeight of a button is incorrect if button has an icon
13575 Dictionary - iphoneSetKeyboardReturnKey examples
12108 Better argument handling for livecode server scripts
11569 Scrolling a group crashes LiveCode when hiding scrollbar
selectionChanged not sent on arrow navigation

Specific bug fixes (7.1.0-dp-1_release)

15572  French application menu is not translated.
11866  iphoneHeadingCalibrationTimeout returns empty

Specific bug fixes (7.1.0-dp-1)

15577  Execution error caused when trying to set the "visited" property of a text link
15571  textEncode crashes with encoding UTF-32
15570  Do not run "after" handlers if an error occurs.
15569  Referenced jpg crashes application on Linux
15568  Crash if SSL lib can't load properly - ARTS
15566  Preferences Mobile Support can't find the SDK for Xcode 6.4
15561  Tooltip appears for controls out of the visible window
15556  Mouse focus not synced when object is created
15511  Erroneous assertion failure logged on server when doing 'go stack'
15508  Allow debug mode standalone engines to load script-only TEST_STACKs
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
15494  mobileControlget ID
15472  Invisible player appears and causes LC to hang
15457  Repeat for each element subtly different in 7.0
15439  enabledTracks "can't set this property" in LC 7
15321  "record sound" is not creating a file in LC7 (Windows)
15263  Large files can EOF prematurely on Linux.
15259  DataGrid library missing in standalones
15191  Windows - Hiding a player in Run mode does not update the screen
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 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 backlight property
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

IDE changes

Specific bug fixes (7.1.2-rc-1)
16735  Added beta entitlement for testflight
16304  Visual effect broken on textChanged
16052  Update default font to San Fancisco for MacOSX 10.11

Specific bug fixes (7.1.1-rc-1)
16079  breakpoints & breaks do not work until after an answer issues
16058  Business edition is seen as Community edition by the Standalone Settings stack
15918  create/clone/copy object and setting position to negative does not work
14502  Return key doesn't trigger default button in Answer dialog
14465  Rulers don't stay fixed to the stack when moving the stack
14224  Combo box will not allow typing into an "Ask" dialog.
13826  Newly created objects are forced to appear on-screen
13709  some message box output in LC when aligning text using menu Text->Align->...
13646  graphic bug in standalone application settings window
13588  "at" option for "edit script" command not honored
11872  impossible to change the case from all caps etc
10267  Add Hardware accelerated option for Android Standalone Applications
3962   Delete key completely clear Ask dialog

Specific bug fixes (7.1.0-rc-2)
15723  Incorrect wording in Business edition activation screen
15717  iOS Version resets to 5.1.1

Specific bug fixes (7.1.0-dp-1)
15580  Mobile Support JDK path not updated after changing it
15545  Setting spaceAbove or spaceBelow on the first line of a table breaks it
15535  Magnify palette cannot be used in a standalone
15522  position of cell editing field in basic table field
15515  Find and replace does not support arrow keys
15510  Project Browser forces the unlocking of the cursor
15478  message box does not auto-scroll
6649   Editing a table field cell shows focus at the wrong place when margins are changed.
LiveCode Builder changes
Previous Release Notes

7.1.2 Release Notes
http://downloads.livecode.com/livecode/7_1_2/LiveCodeNotes-7_1_2.pdf

7.1.1 Release Notes
http://downloads.livecode.com/livecode/7_1_1/LiveCodeNotes-7_1_1.pdf

7.1.0 Release Notes
http://downloads.livecode.com/livecode/7_1_0/LiveCodeNotes-7_1_0.pdf

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

6.7.9 Release Notes

6.7.8 Release Notes

6.7.7 Release Notes

6.7.6 Release Notes

6.7.4 Release Notes

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

6.6.2 Release Notes
http://downloads.livecode.com/livecode/6_6_2/LiveCodeNotes-6_6_2.pdf

6.6.1 Release Notes
http://downloads.livecode.com/livecode/6_6_1/LiveCodeNotes-6_6_1.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