LiveCode 8.1.10 Release Notes

- Overview
- Known issues
- Platform support

  - Windows
  - Linux
  - Mac
  - iOS
  - Android
  - HTML5

- Setup

  - Installation
  - Uninstallation
  - Reporting installer issues
  - Activating LiveCode Indy or Business edition
  - Command-line installation
  - Command-line uninstallation
  - Command-line activation for LiveCode Indy or Business edition

- Engine changes

  - Improved ul_TraceLocals output (8.1.10-rc-1)
  - Script-only deploy (8.1.10-rc-1)
  - Script only stacks with behavior (8.1.10-rc-1)
  - Send script form of send command (8.1.10-rc-1)
  - Add param that suppresses success message when building standalone (8.1.10-rc-1)
  - Throw error when changing behavior from behavior script (8.1.10-rc-1)
  - Fix problems with printing PDFs to some printers (8.1.10-rc-1)
  - Ensure CMYK JPEGs display correctly on Mac (8.1.10-rc-1)
  - Add builtin implementation of field 'Select All' (8.1.10-rc-1)
  - Improve efficiency of compiled regex cache (8.1.10-rc-1)
  - Ensure matchText and replaceText don't affect target string (8.1.10-rc-1)
  - clipboard always converts plain text to styled text (8.1.10-rc-1)
  - Fix deletion of the target in safe cases (8.1.10-rc-1)
  - Type should work with accented characters (8.1.10-rc-1)
  - Ensure browser widgets are in the correct location (8.1.10-rc-1)
  - Synthesize an MS Paint compatible clipboard format for images (8.1.10-rc-1)
  - revAvailableHandlers and revAvailableVariables now in all engines (8.1.10-rc-1)
  - Fix treatment of NUL containing arguments in ask dialogs (8.1.10-rc-1)
  - Remove legacy mergExt externals (8.1.10-rc-1)
  - Make the encoding property of field char chunks more useful (8.1.10-rc-1)
  - Support resizing stacks for orientation changes in fullscreen modes (8.1.10-rc-1)
  - Improve export/import snapshot from screen on iOS (8.1.10-rc-1)
  - Specific engine bug fixes (8.1.10-rc-2)
  - Specific engine bug fixes (8.1.7-rc-1)
  - Specific engine bug fixes (8.1.10-rc-1)

- IDE changes

    - Create script only stack behavior (8.1.10-rc-1)
    - Default handlers (8.1.10-rc-1)
    - First run backdrop (8.1.10-rc-1)
    - ideScriptEdited message (8.1.10-rc-1)
    - Script editor handler menu (8.1.10-rc-1)
    - Interactive Tutorial syntax (8.1.10-rc-1)
    - Allow substack to become a mainstack via property inspector (8.1.10-rc-1)
    - Reinstate store tab of extension manager (8.1.10-rc-1)
    - Drag and drop stackfiles (8.1.10-rc-1)
    - Specific IDE bug fixes (8.1.10-rc-1)

- LiveCode Builder changes

    - LiveCode Builder Standard Library
    - Specific LCB bug fixes (8.1.10-rc-1)

- LiveCode extension changes

    - Tree View widget
    - Specific extension bug fixes (8.1.10-rc-1)

- Dictionary additions
- Previous release notes

Overview

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

- LiveCode Indy and Business editions now come with the tsNet external, which supercharges LiveCode's Internet features and performance. LiveCode 8.1 also introduces mergHealthKit, for accessing activity, sport and health data on iOS devices.

- The standalone builder now has a greatly-improved user experience for including externals, script libraries and LiveCode Builder extensions in your cross-platform application. Usually, it'll now do the right thing automatically, but you can still select the specific inclusions you need.

- The IDE has lots of other upgrades, too: a keyboard-navigable Project Browser that highlights any scripts that failed to compile, an improved dictionary user interface, and access to the message box just by starting to type.

- The player control can be used in Windows application without any need for users to install any additional libraries or dependencies, thanks to a brand new player implementation based on DirectShow. For most apps, it should now be unnecessary to install or use QuickTime at all.

- The LiveCode Builder programming language has had some enhancements as part of the

2

Infinite LiveCode project. Variables now get initialised by default, `unsafe` blocks and handlers can be used to flag sections of code that do dangerous things, and you can even include raw `bytecode` if necessary.

Known issues

- The installer will currently fail if you run it from a network share on Windows. Please copy the installer to a local disk before launching on this platform.

- The browser widget does not work on 32-bit Linux.

- 64-bit standalones for Mac OS X do not have support for audio recording.

Platform support

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

Windows

LiveCode supports the following versions of Windows:

- Windows XP SP2 and above
- Windows Server 2003
- Windows Vista SP1 and above (both 32-bit and 64-bit)
- Windows 7 (both 32-bit and 64-bit)
- Windows Server 2008
- Windows 8.x (Desktop)
- Windows 10

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

Linux

LiveCode supports the following Linux distributions, on 32-bit or 64-bit Intel/AMD or compatible processors:

- Ubuntu 14.04 and 16.04
- Fedora 23 & 24
- Debian 7 (Wheezy) and 8 (Jessie) [server]
- CentOS 7 [server]

LiveCode may also run on Linux installations which meet the following requirements:

- Required dependencies for core functionality:

    - glibc 2.13 or later
    - glib 2.0 or later

- Optional requirements for GUI functionality:

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

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

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

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

Mac

The Mac engine supports:

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

iOS

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

Currently, the supported versions of Xcode are:

- Xcode 4.6 on MacOS X 10.7
- Xcode 5.1 on MacOS X 10.8
- Xcode 6.2 on MacOS X 10.9
- Xcode 6.2 and 7.2 on Mac OS X 10.10
- Xcode 8.2 on MacOS X 10.11
- Xcode 9.2 on MacOS 10.12 (Note: You need to upgrade to 10.12.6)
- Xcode 9.3 on MacOS 10.13 (Note: You need to upgrade to 10.13.2)

It is also possible to set other versions of Xcode, to allow testing on a wider range of iOS simulators. For instance, on OS X 10.10 (Yosemite), you can add *Xcode 5.1* in the *Mobile Support* preferences, to let you test your stack on the *iOS Simulator 7.1*.

We currently support deployment for the following versions of iOS:

- 6.1 [simulator]
- 7.1 [simulator]
- 8.2 [simulator]
- 9.2
- 10.2
- 11.2
- 11.3

Android

LiveCode allows you to save your stack as an Android application, and also to deploy it on an Android device or simulator from the IDE.

Android deployment is possible from Windows, Linux and Mac OSX.

The Android engine supports devices using ARMv6, ARMv7 or ARMv8 processors. It will run on the following versions of Android:

- 2.3.3-2.3.7 (Gingerbread)
- 4.0 (Ice Cream Sandwich)
- 4.1-4.3 (Jelly Bean)
- 4.4 (KitKat)
- 5.0-5.1 (Lollipop)
- 6.0 (Marshmallow)
- 7.0 (Nougat)
- 8.0 (Oreo)

To enable deployment to Android devices, you need to download the Android SDK, and then use the 'Android SDK Manager' to install:

- the latest "Android SDK Tools"
- the latest "Android SDK Platform Tools"

You also need to install the Java Development Kit (JDK). On Linux, this usually packaged as "openjdk". LiveCode requires JDK version 1.6 or later.

Once you have set the path of your Android SDK in the "Mobile Support" section of the LiveCode IDE's preferences, you can deploy your stack to Android devices.

Some users have reported successful Android Watch deployment, but it is not officially supported.

HTML5

LiveCode applications can be deployed to run in a web browser, by running the LiveCode engine in JavaScript and using modern HTML5 JavaScript APIs.

HTML5 deployment does not require any additional development tools to be installed.

LiveCode HTML5 standalone applications are currently supported for running in recent versions of Mozilla Firefox, Google Chrome or Safari. For more information, please see the "HTML5 Deployment" guide in the LiveCode IDE.

Setup

Installation

Each version of LiveCode installs can be installed to its own, separate folder. This allow multiple versions of LiveCode to be installed side-by-side. On Windows (and Linux), each version of LiveCode has its own Start Menu (or application menu) entry. On Mac OS X, each version has its own app bundle.

On Mac OS X, install LiveCode by mounting the `.dmg` file and dragging the app bundle to the `Applications` folder (or any other suitable location).

For Windows and Linux, the default installation locations when installing for "All Users" are:

Platform	Path
Windows	`<x86 program files folder>/RunRev/LiveCode <version>`
Linux	`/opt/livecode/livecode-<version>`

The installations when installing for "This User" are:

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

Note: If installing for "All Users" on Linux, either the **gksu** tool must be available, or you must manually run the LiveCode installer executable as root (e.g. using **sudo** or **su**).

Uninstallation

On Windows, the installer hooks into the standard Windows uninstall mechanism. This is accessible from the "Add or Remove Programs" applet in the windows Control Panel.

On Mac OS X, drag the app bundle to the Trash.

On Linux, LiveCode can be removed using the `setup.x86` or `setup.x86_64` program located in LiveCode's installation directory.

Reporting installer issues

If you find that the installer fails to work for you then please report it using the LiveCode Quality Control Centre or by emailing support@livecode.com.

Please include the following information in your report:

- Your platform and operating system version
- The location of your home or user folder
- The type of user account you are using (guest, restricted, admin etc.)
- The installer log file.

The installer log file can be located as follows:

Platform	Path
Windows 2000/XP	`<documents and settings folder>/<user>/Local Settings/`
Windows Vista/7	`<users folder>/<user>/AppData/Local/RunRev/Logs`
Linux	`<home>/.runrev/logs`

Activating LiveCode Indy or Business edition

The licensing system ties your product licenses to a customer account system, meaning that you no longer have to worry about finding a license key after installing a new copy of LiveCode. Instead, you simply have to enter your email address and password that has been registered with our customer account system and your license key will be retrieved automatically.

Alternatively it is possible to activate the product via the use of a specially encrypted license file. These will be available for download from the customer center after logging into your account. This method will allow the product to be installed on machines that do not have access to the internet.

Command-line installation

It is possible to invoke the installer from the command-line on Linux and Windows. When doing command-line installation, no GUI will be displayed. The installation process is controlled by arguments passed to the installer.

Run the installer using a command in the form:

```
<installer> install noui [OPTION ...]
```

where `<installer>` should be replaced with the path of the installer executable or app (inside the DMG) that has been downloaded. The result of the installation operation will be written to the console.

The installer understands any of the following `OPTION`s:

Option	Description
`-allusers`	Install the IDE for "All Users". If not specified, LiveCode will be installed for the current user only.

- **Option** `desktopshortcut`	Place a shortcut on the Desktop (Windows-only) **Description**
`-startmenu`	Place shortcuts in the Start Menu (Windows-only)
`-location LOCATION`	The folder to install into. If not specified, the `LOCATION` defaults to those described in the "Installation" section above.
`-log LOGFILE`	The file to which to log installation actions. If not specified, no log is generated.

Note: the command-line installer does not do any authentication. When installing for "All Users", you will need to run the installer command as an administrator.

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

On Windows, the command is:

```
start /wait <installer> install noui [OPTION ...]
```

Command-line uninstallation

It is possible to uninstall LiveCode from the command-line on Windows and Linux. When doing command-line uninstallation, no GUI will be displayed.

Run the uninstaller using a command of the form:

```
<uninstaller> uninstall noui
```

Where is *.setup.exe* on Windows, and *.setup.x86* on Linux. This executable, for both of the platforms, is located in the folder where LiveCode is installed.

The result of the uninstallation operation will be written to the console.

Note: the command-line uninstaller does not do any authentication. When removing a version of LiveCode installed for "All Users", you will need to run the uninstaller command as an administrator.

Command-line activation for LiveCode Indy or Business edition

It is possible to activate an installation of LiveCode for all users by using the command-line. When performing command-line activation, no GUI is displayed. Activation is controlled by passing command-line arguments to LiveCode.

Activate LiveCode using a command of the form:

```
<livecode> activate -file LICENSEFILE -passphrase SECRET
```

where `<livecode>` should be replaced with the path to the LiveCode executable or app that has been previously installed.

This loads license information from the manual activation file `LICENSEFILE`, decrypts it using the given `SECRET` passphrase, and installs a license file for all users of the computer. Manual activation files can be downloaded from the My Products page in the LiveCode account management site.

It is also possible to deactivate LiveCode with:

```
<livecode> deactivate
```

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

On Windows, the command is:

```
start /wait <livecode> activate -file LICENSE -passphrase SECRET
start /wait <livecode> deactivate
```

On Mac OS X, you need to do:

```
<livecode>/Contents/MacOS/LiveCode activate -file LICENSE -passphrase SECRET
<livecode>/Contents/MacOS/LiveCode deactivate
```

Engine changes

Improved ul_TraceLocals output (8.1.10-rc-1)

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

Script-only deploy (8.1.10-rc-1)

It is now possible to use script-only stacks in the mainstack and auxiliary stack parameters to the deploy command.

Script only stacks with behavior (8.1.10-rc-1)

Script only stacks can now store a stack behavior as part of the file format. A 'with behavior' clause is added to the header of a script only stack, if it has a behavior property which references a stack.

When a script-only-stack with such a clause is loaded, the behavior is set as part of the loading.

Send script form of send command (8.1.10-rc-1)

The syntax

```
send script <script> to <obj>
```

has been added to allow a chunk of script to be executed in the context of an object without any attempted evaluation of parameters that occurs with the original form of the send command.

For example, suppose there is a stack named "Stack" with script

```
on doAnswer pParam
    answer pParam
end doAnswer

function myName
    return the short name of me
end myName
```

and a button on the stack named "Button" with script

```
on mouseUp
    send "doAnswer myName()" to this stack
    send script "doAnswer myName()" to this stack
end mouseUp

function myName
    return the short name of me
end myName
```

clicking the button would result in an answer dialog first saying "Button" as the `myName` function would be evaluated in the button context, then "Stack" as using the `script` form would result in the `myName` function being evaluated in the stack context.

Add param that suppresses success message when building standalone (8.1.10-rc-1)

By default `revSaveAsStandalone` displays the message
`answer information "Standalone application saved successfully."` when it is done. You can turn off this message by setting the test environment to true but doing so suppresses all error messages and other feedback as well.

I am calling `revSaveAsStandalone` from my own scripts multiple times and want feedback and error reporting but not the success message. Adding an additional parameter to `revSaveAsStandalone` that suppress the success message would allow this.

Throw error when changing behavior from behavior script (8.1.10-rc-1)

Previously it was theoretically possible to change the behavior of an object from that object's existing behavior script. This will now result in an execution error

```
parentScript: can't change parent while parent script is executing
```

This change was necessarily as the engine would occasionally crash when changing a behavior this way, and would be guaranteed to crash if stepping over the behavior script line that changes the behavior.

Fix problems with printing PDFs to some printers (8.1.10-rc-1)

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

Ensure CMYK JPEGs display correctly on Mac (8.1.10-rc-1)

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

Add builtin implementation of field 'Select All' (8.1.10-rc-1)

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

Improve efficiency of compiled regex cache (8.1.10-rc-1)

The efficiency of lookups of previously compiled regexs has been improved. To take full advantage of the regex cache make sure that you either use a string constant for the regex pattern, or a variable which is not mutated between uses. e.g.

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

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

Ensure matchText and replaceText don't affect target string (8.1.10-rc-1)

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

clipboard always converts plain text to styled text (8.1.10-rc-1)

Whenever `text` was placed on the clipboard, it was first converted to LiveCode styled text and then put on the clipboard as styled text, RTF, HTML, and plain text. This introduced errors when pasting to other applications since they would prefer the HTML version which made the text appear double spaced.

Fix deletion of the target in safe cases (8.1.10-rc-1)

You can now safely 'delete the target' as long as there are no handlers on the stack owned by the target.

After deleting 'the target', 'the target' will become empty which will result in an execution error when an attempt is made to dereference it.

Type should work with accented characters (8.1.10-rc-1)

The `type` command now handles Unicode characters in a manner consistent with normal keyboard entry. If a Unicode character is typed and it has a native mapping, then it is propagated as a keypress with the keycode being the code of the character. If it has no native mapping, it is propagated with keycode equal to the Unicode codepoint with bit 22 set to 1. In either case the string value of the keypress is the Unicode codepoint.

Ensure browser widgets are in the correct location (8.1.10-rc-1)

Browser widgets in nested groups now remain in the correct location rather than shifting down vertically out of sync with everything else.

Synthesize an MS Paint compatible clipboard format for images (8.1.10-rc-1)

The engine will (once again) synthesize a DIBV5 format when an image is copied to the clipboard. This will be a 32-bit RGBA DIB. Windows then automatically synthesizes a 24-bit RGB DIB format.

revAvailableHandlers and revAvailableVariables now in all engines (8.1.10-rc-1)

The 'the [effective] revAvailableHandlers' and 'the revAvailableVariables' properties are now available in both the IDE and Standalone engines.

Note: These properties are currently undocumented, and reserved for IDE related uses. You are free to use in your own code, but their semantics could change in any release.

Fix treatment of NUL containing arguments in ask dialogs (8.1.10-rc-1)

Prior to 7, any arguments passed to LiveCode provided ask dialogs (e.g. ask question) containing NUL would be truncated at the NUL. After 7, any such arguments would cause incorrect calling of the ask dialog. The pre-7 behavior has been resurrected, meaning that trailing NUL bytes in arguments passed to ask dialogs will be ignored.

Remove legacy mergExt externals (8.1.10-rc-1)

The following mergExt deprecated externals are no longer included in LiveCode.

- mergAES - we have revsecurity based encryption for mobile
- mergDropbox & mergDropboxSync - these use the now abandoned by Dropbox (v1 API). We have a script library available for v2 API.
- mergSocket - we have sockets in the engine for mobile
- mergZXing - no longer supportable as the ZXing project no longer supports iOS. Use mergAVCam for barcode capture instead.

Make the encoding property of field char chunks more useful (8.1.10-rc-1)

The 'encoding' char-level field property will now return native if all chars in the chunk can be encoded in the native encoding, and unicode otherwise.

This means that the property will now return the identical value as it did in 6.7 and before, assuming that the field text hadn't had its encoding changed by script (via the textFont ',unicode' flag).

Support resizing stacks for orientation changes in fullscreen modes (8.1.10-rc-1)

A new mobile command `mobileSetFullScreenRectForOrientations` has been implemented to allow stacks that use the `fullscreenMode` property to be resized when the device orientation changes.

Improve export/import snapshot from screen on iOS (8.1.10-rc-1)

The from screen form of export/import snapshot has been changed to use a different API on iOS7+, which allows a greater variety of native layers to be captured.

Specific engine bug fixes (8.1.10-rc-2)

21228   Ensure android working screenrect is updated when the keyboard is presented

Specific engine bug fixes (8.1.7-rc-1)

17098   Ensure cursor moves to end of last tab in line.

Specific engine bug fixes (8.1.10-rc-1)

10947   Fix hypercard-compatibility dynamic path behavior

11039   Throw error when changing behavior from behavior script

11146   Ensure that the initial orientation is not upside down on Android when "portrait" is selected.

11170   Disable keyboard suggestion when entering password in Android native input field.

11313   Support mobilePickPhoto() maximum width and height settings on Android

11727   Fix arrow key not work in Android field

12187   Make sure keyboardActivated/keyboardDeactivated messages are sent when the status bar is hidden on Android

13055   Improve formatting of try syntax description

13151   Correct example of setting "listIndent" for whole field

13180   Allow GPS access from Android Browser

13370   Correct terminology in "convert" command documentation

13482   Document optional catch clause in try control structure

13857   Correct dictionary entry for revProfile

13935   Fix modal dialog opening behind other windows on Linux

13992   Fixed bug causing crash on mouse enter

14080   Fix find command not finding in specified field when not on current card

14238   Ensure background pattern stays aligned in long fields

14266   Fixing crash when using "import eps"

15302   Fix common misspelling of occurred

15384   Fix incorrect handling of a 204 response from a server.

15997   Implement player 'mirrored' property on Windows

16131   Ensure backdrop is sized to fill the screen on Linux

16388   Fix setting dragdata["files"] when filename has spaces

16551   Ensure the "hilitedButtonName" returns the name of the hilited button

16758   Fixed bug causing crash when setting 65535 points to a graphic

16965   Fix incorrect placement of browser widget after stack rect change when fullscreenmode used.

17008   Fix selection handles remaining after selected object deleted

17247   Remove selection artefacts when handles are drawn outside of parent group rect

17323   Ensure backdrop window is behind all other windows on Linux

17540   Fix crash on Mac when displaying IDE usage message

17541   Fix problems with printing PDFs to some printers

17577   Make sure we can set the hilitedItemName property of the navBar widget

17622   Fix extra data added on Windows when pasting html data from the clipboard

17639	Fix vertical placement of caret on long wrapped lines
17657	Make sure modifier keys are recognised in keysDown()
17779	Fix scrolling group drawing outside its bounds when acceleratedRendering used.
17850	Fix inconsistent browser callbacks in browser widget on iOS
17918	Make sure setting clipboarddata["text"] and clipboarddata["html"] works as expected
17969	Fixed bug preventing correct rendering of ovals when creating them
17973	Make sure `the machine` can distinguish between iOS device and simulator
18058	Fix keyboard not show in landscape orientation
18243	Ensure horizontal two finger scrolling on Linux respects the system settings
18264	Don't fail standalone build completely if unlicensed platforms are selected
18273	Prevent crash when rendering card with invalid objptr
18275	Enable sqlite FTS5 feature
18277	Calculate the height of the mac desktop space correctly
18287	Update sqlite version to 3.15.0
18293	Crash when deleting a stack that is used as a popup menu
18295	Fix empty specialFolderPath("resources") on Windows
18343	Fix incorrect result from itemOffset when first character of stringToSearch is the delimiter
18349	Fix variable contents modified when used to set stack name
18353	Remove duplicated urlResponse documentation
18358	Make sure mobileControlGet does not return rounded values of startTime/endTime/currentTime
18369	Add explicit instruction to DMG images
18379	Don't include incorrectly copied resource forks in standalones
18392	Ensure "ceiling" is listed as a reserved word
18406	Fix delay in triggering handlers when called by JavaScript in browser widget
18407	Fixed bug preventing the use of "set the visible of line <> of field <>"
18440	Respect SB Copy Files pane relative / absolute path distinction on mobile
18441	Make sure the purchaseStateUpdate callback is sent with status=complete when necessary
18443	Ensure 'private' clipboard is reset when another application sets the clipboard
18444	Make sure put cookie with empty value works as expected
18447	Moved misplaced text in iconGravity dictionary entry
18454	Allow socket to send broadcast packet on Android.
18459	Fix incorrect behavior of files() and folders() function on Android.
18472	'load url' is not properly cleaned up on socketError
18473	Prevent hang when adjusting field pixmap offset
18484	Prevent mobilePlaySoundOnChannel crash on Android
18488	Error returned by hostnametoaddress was not being reported in libURL.
18496	Fix memory leak when using filter on unicode strings
18498	Ensure bundled Android externals are available on Windows and Linux
18499	Fix libUrlSetStatusCallback on mobile platforms when tsNet is in use
18514	Make sure setting clipboarddata["html"] works as expected on Windows
18521	Resolve folder path before processing files(folder) and folders(folder)
18526	Allow command key shortcuts to work in color dialog

18536	Added support for Xcode 8.1 / iOS 10.1
18539	Don't change the defaultFolder on startup
18566	libURL inserts "::" between host and port when creating CONNECT request
18578	Ensure color name rgb value mapping is in alphabetical order
18600	Fix crash when quitting from script editor
18604	Fix crash when converting objective-c objects to LiveCode values
18610	Add code examples to tsNet dictionary entries
18614	Fix Linux player crash when accessing properties
18619	Delete slash at the end of specialFolderPath("resources") to be consisten with other result of special folders.
18623	Remove tsneterr: from the start of output of tsNetVersion()
18625	Fix browser javascripthandlers not working on iOS after loading a new page
18626	Make sure the Standalone Application Settings on Windows are respected
18632	Mark the copyResource function as deprecated
18642	Fixed crash on iOS 10 when trying to read local notifications
18652	Fix occasional crash when getting the clipboarddata["text"] on Windows.
18653	Encode bundle display name as utf-8 in ios app plist
18666	Fix crash when find command matches text in sharedText field on non-current card
18670	Fixed bug preventing all table names being retrieved in MySQL db's
18683	Fix crash on iOS 10 when the app needs access to the device's microphone
18686	Fix a crash related to the message watcher
18689	Fixed bug causing erronous cursor movement when using arrow keys
18690	Provide mergExt Builds for building against iOS 10.1 SDK
18691	Provide tsNet Builds for building against iOS 10.1 SDK
18703	Fix handling of new tsNet network requests when an existing network request is still waiting to connect
18709	Cannot deploy an app to iOS 10.1 simulator
18724	Fix incorrect cross-references in lockLocation dictionary entry
18738	Fix data loss when cr inserted into a styledText run
18755	Fix loss of BMP as supported clipboard image format
18762	Fix a rare crash on saving after cloning a field
18778	Send standalone saving messages at correct time
18783	Resolve delay in opening Windows standalones that include tsNet 1.2.4 and the Internet inclusion
18809	Prevent lock up of PI when not selecting choice from font menu
18810	Fix a crash when cutting controls
18812	Fix crash when opening cards referencing images on non-open cards
18823	Fix a crash when popping up transient windows
18824	Fix a crash when drag-selecting controls
18833	Don't change name of tsNet stack during standalone build
18852	Fix exception thrown in IDE when saving standalone with more than one stack
18863	Fix encrypt/decrypt operations broken by the OpenSSL 1.1.0 update
18870	Fix crash caused by calling tsNetGetSync() repeatedly against the same URL
18871	Add support for upper and lower case field shortcuts on Mac
18873	Fix documentation of valid values for lineSize property
18890	Fix crash when playing non-imported audioclip

18893	Fix formatting in description of stack mode property
18894	Don't strip context tags from HTML clipboard formats
18900	Fix a crash when closing a stack with substacks still open
18911	Fix graphical artefacts when reshaping polygon graphics while selected
18912	Ensure objects can't be deleted if their behaviors are handling a message or they are the target
18923	Ensure CMYK JPEGs display correctly on Mac
18925	Prevent crashes on memory exhaustion
18938	Fix text encoding issues when pasting HTML into LiveCode
18939	Fix player image incorrectly scaled in edit mode
18946	Fix browserNavigateComplete not firing when document has frames
18948	Make 'obj of me' consistent across all control types
18955	Fix crash when using HTML file input dialog in browser widget
18958	Make sure our prebuilt libs do not use reserved (by Apple private APIs) function names
18959	Fix crash after getting the points of a regular polygon graphic
18962	Fix crash when using field
18969	Add builtin implementation of field 'Select All'
18977	Ensure setting the card triggers openCard / closeCard in no UI mode
18982	Improve shift+click behavior of text selection in XPDF
18988	Deal with stack id cache correctly on stack delete
19003	Stackfiles under the mainstack path preserve folder structure in standalone
19004	Improve efficiency of compiled regex cache
19005	Ensure matchText and replaceText don't affect target string
19008	Place modals at their own location when neither the defaultStack or topStack can provide useful relative locations
19009	mergExt external builds for iOS 10.2
19010	Provide iOS 10.2 builds for tsNet
19017	Fix crash when deleting object which is being moved
19024	Prevent crash when calling `mobileStoreRequestProductDetails`
19026	Fix DNS resolver issue causing connection hang when using tsNet on Linux
19032	Prevent crash when creating and deleting two data grids
19035	Fix crash when using tsNet with OS X 10.6 and 10.7
19036	Fix incorrect local file path when setting url of browser widget on Android.
19044	Prevent crash when deleting stack which has an open substack
19045	Fix BiDi algorithm for surrogate pairs
19049	Prevent crash when deleting mousestack stack via button click
19051	Make sure the player endtime property cannot be set to a value greater than the player duration
19060	Ensure error when binding widget is caught correctly
19069	Fixed bug causing crash when building standalone
19076	Prevent redraw recursion when going to stack twice in no ui mode
19080	Do not show linking warnings when building iOS standalones
19084	Ensure text is copied as unstyled on mac
19088	Do not show "Success" dialog if an error occurred in S/B
19093	Prevent recursion when fetching mouseColor in mouseMove handler

19097	Ensure all menus are updated after mouseDown to menu group
19105	Resolve crash when deleting a stack in the message path in a front script
19108	Correct references for scriptOnly property docs
19116	Ensure tsNetGetStatus reports transfer status as "uploading" appropriately during POST requests
19120	Ensure substacks with cantDelete don't cause execution errors when deleting the mainstack
19121	Pass correct folder to standaloneSaved message
19138	Update OpenSSL to version 1.1.0d
19154	Fix widget browser stuck on handling javascript
19158	Prevent crash when using undo to bring back removed controls
19185	Ensure the dragData["private"] isn't cleared during a drag
19192	Add ios and android to "allowDatagramBroadcasts" dictionary entry
19200	Make sure printSettings are set correctly
19206	clipboard always converts plain text to styled text
19212	Prevent a crash when calling mobileComposeMail with just one param (tSubject)
19215	Make sure botton icons are present in standalones when building for multiple platforms
19229	Fix crash when connecting an IR Receiver
19246	Remove objects from message path ASAP when deleted
19275	Fix Android crash when a stack is deleted shortly after switching to another stack.
19279	Prevent mobilePickMedia crash without 'Write External Storage' permission
19287	Make sure the clickLoc is updated on mouseDown/touchEnd on mobile
19293	Server returns 'ELF' over HTTP
19298	Make sure "Search for Inclusions" detects correctly widget inclusions on iOS simulator
19307	Prevent crash when saving standalone while player is playing
19313	Fix crash when saving field with fdata
19320	Fix SVG parsing of 'a' instruction
19327	Fix incorrect browser widget location when stack has a menu on OSX
19328	Standalone startup stack needs to be mainstack before resolving parent scripts
19352	Fix crash when getting the urlResponse
19361	Added missing parantheses for revIsSpeaking()
19390	Reposition native layer controls correctly when resizing stacks
19404	Fix crash on iOS when calling `play empty` followed by `play path/to/audio/file`
19416	Ensure all object messages are cleared when obj or ancestor is deleted
19417	Use correct comment syntax in docs on documentation
19420	Fix crash on startup when resuming android app after quit
19424	Make sure getting `the securityPermissions` returns the expected result
19457	Prevent crash when deleting selected objects with the backspace key
19469	Make sure the long time format is correct if `twelveHourTime` is false
19490	Make sure launch url sets the result to empty on success
19500	Fix crash when invalid url is used with iOS native browser
19509	Make sure Android Studio works with LiveCode
19515	Ensure the formattedRect of line N always returns the correct result
19520	Make sure mirrored property works correctly on Mac

19525 Fix SFTP connection hang when remote directory doesn't exist

19528 Allow 'relaunch' message to be in behavior of main stack

19529 Fix incorrect snapshot area when primary screen origin is not 0,0

19535 Fixed error causing blue rectangle to be drawn incorrectly when doing CMD-A on text

19538 Fixed bug preventing users from selecting text of length 65535

19541 Fix clipboard ownership checks on windows causing private clipboard data to be cleared

19573 Provide iOS 10.3 builds for tsNet

19577 Fix crashes related to card deletion

19578 Fix crash after leaving edit group mode

19580 Ensure colors round-trip through styledText correctly

19581 Do not send pushNotificationReceived/localNotificationReceived message twice

19584 Allow deletion of a message's target object in a frontscript

19587 Fix deletion of the target in safe cases

19592 Prevent crash when deleting combo box

19593 Type should work with accented characters

19599 Ensure correct source rect is used for 'print card from lt to rb'

19609 Make sure unicode characters display correctly when set htmlText in browser

19613 Make sure setting the htmlText of a field does not remove superscripts

19615 Fixed references in the quit dictionary entry

19620 Update cursor when entering window on windows

19630 Make sure setting the iphoneSetAudioCategory is respected

19633 Modify tsNet libUrl wrapper to treat HTTP status codes > 400 as errors

19635 Ensure browser widgets are in the correct location

19637 Errors building iOS standalones should only be reported once

19646 Make sure using mobile camera does not change the value of mobileLockIdleTimer

19649 Correctly parse multiple bytes escaped as hex in the format function

19650 Fix tsNet proxy support on Linux, OS X and Android

19652 Refresh player on windows when mirroring property is set

19666 Make sure you can set more than one javascriptHandlers on CEF Browser

19668 Fixed bug causing crash when using custom undo

19672 Prevent crash on throwing certain errors

19687 Preserve error from chunk-of-code send form

19688 Ensure 'put the objProp' causes a parse error

19693 Ensure closed players use no system resources

19699 Fixed bug that overrides previous matches in matchText

19701 Try HTTP basic authentication if a HTTP server responds with 401 without supplying the WWW-Authenticate header

19713 Synthesize an MS Paint compatible clipboard format for images

19730 Fix resolution of relative paths of images in Mac standalones

19742 Fix crash when deleting an object when a socket has a reference to a deleted object

19743 Fix crash when checking a watched variable on a deleted object

19766 Fix FTPS connection support on LC server under Linux

19775 Fix various error inconsistencies in selected object cut and delete

19796 Fix crash when SSL is unavailable on HTML5

19797 Implement put before msg in HTML5

19808	Notify property listener when text of control is changed via put cmd
19820	Given the dictionary entries for remove and place references to each other.
19826	Fix iOS simulator deployment with Xcode 8.3.3
19830	Fixed regression introduced by bugfix-19535
19837	Fix crash due to deletion of object with pending message
19841	Message box does not find handlers in behaviors
19843	revAvailableHandlers and revAvailableVariables now in all engines
19857	Make sure `clipsToRect` is included in the group properties
19861	Fix crash on startup on Linux when RGBA is not supported
19869	The effective revAvailableHandlers only includes private handlers of the target
19891	Ensure player controller thumb shows within the allowed range
19893	Ensure player respects startTime in reverse playback
19929	Fix crash when using click command with invalid mouse stack or click stack
19935	Made various improvements to the show and hide dictionary entries
19936	Ensure window masks with no transparency still work on Mac
19950	Fix crash due to invalid object in event queue
19965	Fix error when building a standalone if the added stackfiles have substacks
19972	Make sure setting playRate to negative does have an effect when player has reached the end of movie
19983	Ensure `put URL tUrl` does not return empty when tUrl is invalid
19994	Ensure startup stack substacks are cleanly removed from memory
20014	Fixed several errors in the start/stop using font dictionary entries.
20019	Ensure `GetPixelHeightOfCanvas()` returns the height of the canvas
20030	Ensure the S/B always uses a valid certificate when codesigning iOS standalones
20061	Ensure data is not lost when opening and saving a stack with a widget that is not loaded
20080	Fix malformed documentation of the `the universal time`
20144	Corrected hiliteChanged dictionary entry for the switch button extension.
20167	Fix capitalization of menu item "Hide Others" on Mac
20201	Ensure `the controlNames` does not return numbers instead of names for controls in groups
20206	Mention in `revBrowserOpenCEF` dictionary entry that it is no longer supported on Mac in LC 8+
20209	Generate correct RGB values in rtftext
20212	Ensure setting the enabledTracks of a player is reliable
20231	Fix crash on Windows when exiting with taskbar hidden.
20232	Ensure dragdata["files"] returns a Unix path on all platforms
20239	Ensure answer folder shows the prompt on OSX 10.11 and above
20256	Ensure iOS picker subview width scales correctly
20259	Fix crash when inserting large binary data to SQLite databases that aren't opened with the "binary" option
20269	Throw parse error when dispatch ... with has empty params
20282	Ensure 'the engine folder' returns a LiveCode path on Windows
20285	Fix crash when dispatching to an object and the defaultStack has been deleted
	Fixed bug causing crash when using chunks of the type 'the last char to -4 of

20293	"fdwbfdf"'
20298	Ensure a diamond checkmark is used when requested on Mac
20308	Fix iOS 64-bit Mach-O structure
20310	Make pasting from MS Paint work
20321	Fix treatment of NUL containing arguments in ask dialogs
20323	Remove legacy mergExt externals
20324	Convert dropped file paths correctly on Windows
20335	Mac folder dialog missing "add folder" button
20349	Fix crash when deleting the focused object
20363	Update DIB format when image placed on Windows clipboard
20405	Fixed bug causing incorrect formattedWidth to be returned for buttons
20413	Add documentation for android specialFolderPath external prefix
20419	Fix accelRender issues on Android
20423	Add tsNet builds for iOS SDK 11.0
20424	Add mergExt builds for iOS SDK 11.0
20434	Fix crash on iOS when the app uses HealthKit
20467	Fixed bug causing crash when using multiple players.
20478	Prevent crash on quit when using the commandName
20489	Remove revVideoGrabber external from IDE as it can no longer be supported
20490	Fix crash when closing color dialog on macOS High Sierra
20503	Fix quote key not working with Turkish keyboard layout on Mac
20507	Ensure Y is respected in 'read from socket X for Y'
20510	Fix crash on Windows when using revDatabaseTableNames()
20538	Ensure flushEvents("all") works on MacOS
20565	Fix setting stack to fullscreen hides all other stacks on Linux
20582	Ensure the iOS device plist has correct values for the version of Xcode and SDKs used to build the standalone
20592	Ensure iOS standalones are treated as unique by fingerprint scanning
20627	Set default timeout in tsNet to prevent app hangs when Internet connection drops
20628	Add tsNet builds for iOS SDK 11.1
20633	Ensure vtab doesn't interfere with styling
20641	Added support for splash screens and icon for iPhone X
20642	Fix crash when undoing a group deletion
20654	Fix crash when trying to write to a disconnected socket
20670	[Xcode 9+] Detect correctly if a simulator device is already open
20742	Add tsNet builds for iOS SDK 11.2
20755	Fix crash when calling iPhoneSetRemoteControlDisplay
20758	Fix performance regression in replaceText
20759	Fix crash saving images to iOS photo library
20760	Fixed documentation for tsNetSetTimeouts to specify correct units for all parameters
20763	Fix crash when deleting datagrid then undoing
20811	Make the encoding property of field char chunks more useful
20884	Fixed AppStore submission error when minimum deployment target is iOS 11.0 or more
20898	Fix crash when converting from utf16 with revDataFromQuery

20907	Ensure params passed to iphoneSetDoNotBackupFile are taken into account
20925	Fix crash when repeatedly dragging
20946	Remove 32 bit slice from Mac externals if the standalone supports 64 bit only
20952	Fix shell on Windows Server
20961	Clear the menu object when moving from one menu to another
20986	Ensure mobileSoundOnChannel() returns correct value on iOS
20997	Add tsNet builds for iOS SDK 11.3
21016	Add support for local storage to Android browser
21019	Support resizing stacks for orientation changes in fullscreen modes
21062	Fix detection of HTTP digest authentication with tsNet
21109	Ensure when setting `the fullClipboardData["text"]` to only clear the clipboard if it contains styled text and do not clear private data
6506	Fix regression to watching global variables
8274	Improve modal placement documentation
9092	Fixed typo in the revPrintField dictionary entry.
9992	Improve export/import snapshot from screen on iOS

IDE changes

Create script only stack behavior (8.1.10-rc-1)

The menu for assigning a behavior to a control has two additional options:

- Create behavior from new script only stack
- Create behavior using control script and script only stack Either option will prompt you for a stack name and a location for the script only stack. The new stack will be saved, assigned as the behavior of the control, and then added to the stackfiles property of control's stack.

Default handlers (8.1.10-rc-1)

Objects no longer have default scripts that appear in the script editor when their empty scripts are edited. Instead, all the associated message handlers for the object type now appear in a list underneath the list of handlers that are present in the script. When clicked, these lines add the selected default handler to the end of the current script.

If there is a default script for this handler and object type
in the appropriate location (Toolset/resources/supporting_files/default_scripts/ for 'classic' objects, /support/ for widgets), the content of the handler is obtained from that script (including preceding comments)

If there is no default script for this handler and object type, the handler is constructed using information from the documentation, namely the Summary element is used as a preceding comment to describe the handler, and then the handler declared with all the specified parameters.

First run backdrop (8.1.10-rc-1)

The IDE now has a backdrop by default on first-run. This can be turned off as usual via the view menu. Users with existing preferences should be unaffected.

ideScriptEdited message (8.1.10-rc-1)

A new IDE message has been added:

```
ideScriptEdited pScript, pObj
```

This message is sent when the script of an object as displayed in the script editor is changed. pScript contains the current contents of the script editor field for pObj, which, until applied, is not necessarily the same as `the script of pObj`.

Script editor handler menu (8.1.10-rc-1)

The Handler menu of the script editor menubar has been modified in accordance with the default handler changes to the script editor handler list. It now has the following structure:

Go to handler... -> list of extant handlers Add default handler... -> list of default handlers Show default handlers

The show default handlers menu item toggles the script editor preference to show the default handler list, which defaults to true.

Interactive Tutorial syntax (8.1.10-rc-1)

The syntax `load stack <FileName>` has been added to interactive tutorials. This allows prepared stacks to be imported as operating stacks in the current tutorial.

The prepared stack will be loaded from the internal resources folder of the tutorial (i.e. from `_resources/<FileName>`). Any `cTutorialTag` custom property of objects on the stack will be converted to tags for objects which can subsequently be used in the current tutorial.

Allow substack to become a mainstack via property inspector (8.1.10-rc-1)

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

Reinstate store tab of extension manager (8.1.10-rc-1)

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

Drag and drop stackfiles (8.1.10-rc-1)

You can now drag and drop stack files onto the stackFiles field in the PI.

Specific IDE bug fixes (8.1.10-rc-1)

15917	Font Size for Project Browser can now be set from LiveCode Preferences -> Project Browser
17447	Reinstate resize checkbox in property inspector Position pane
17448	Make sure messages are sent when going to stacks/cards from the Project Browser
17485	Allow accessing Image Library and Object Library from LiveCode menubar
17536	Fix or mitigate effect of nudging many controls at once
17618	Reinstate 'fit content' button in property inspector Position pane
17819	Enable cmd+c in dictionary
17889	Repaired confusing layout of fill gradient control in Property Inspector
18029	Show 20 fonts at a time in property inspector font menu
18035	Make sure the gradient popup stack is displayed as expected
18088	Allow setting multi-line tooltips from the Property Inspector
18177	Reinstate text properties and graphic effects to datagrid inspector
18201	Make sure rulers can be hidden
18290	Use datagrid template safely while building custom headers
18292	Fix hidden palettes not reappearing
18300	property inspector custom property list is not sorted
18302	Retain custom prop changes when clicking on tree view in editor
18393	[Project Browser] Change "Sort controls by number" to "Sort controls by layer" to avoid confusion + make sure they are sorted numerically
18452	Saving a substack from the Project Browser no longer asks for a path to save
18455	Show the correct version of LiveCode in Start Center title
18460	Mark stack as edited when property changed from the PI
18483	Prevent iOS display name standalone setting becoming utf-8 encoded data
18485	Ensure relayering menu items don't relayer objects out of owner groups
18491	Allow substack to become a mainstack via property inspector
18549	Make sure `lock cursor` works in the IDE
18557	Ensure dragging object from tools palette is smooth
18586	Make sure the Project Browser stack/card/group view can always expand
18598	Prevent error on backspace in empty script editor
18685	Remember the last position of menubar on Windows and Linux
18701	Prevent over-enthusiastic save prompts
18721	Make sure unchecking "Notify me of development releases" is respected
18726	Make sure the Bug Report checkmark appears at the correct place in Standalone Settings
18791	Fix PI list editors not updating when value changed
18804	Update locked inspectors when mainstack names change
18857	Import as control > Text file doesn't set text of control
18878	Setting stackFiles in PI causes an error if you "cancel" the file dialog or select multiple

files

18897	"Show Sections" for Project Browser can now be set from LiveCode Preferences -> Project Browser
18915	Allow a 'set the name of stack' step in interactive tutorials
18930	Reinstate store tab of extension manager
18981	Added tooltip to iOS icon and splash screen selection
18987	Reinstate fixedLineHeight for tableField
18991	Disable custom property editor when no node selected
19015	Reset the templateStack after datagrid creation
19152	Show warning if the new stack name begins with "rev"
19153	Ensure objects can not be dragged to invisible open stacks from the tools palette
19160	Make sure the S/B respects the "iPad initial orientations" settings
19177	Update guide images for LiveCode 8
19178	Add test to ensure default folder doesn't change when loading IDE
19179	Add tests for standalone builder inclusions
19181	Ensure tutorial has location set when being skipped
19188	Make outputting debug vars from message box work in all contexts
19195	Allow vertical scrolling in "Value" field of Variable Visualizer window
19196	Ensure extension is installed before deleting files
19264	Ensure LCB errors display reasonably in script editor
19439	Ensure the "Effects" settings stack always appears onscreen
19451	Don't try to delete breakpoint while it is a target in the call stack
19480	Ensure message box execution succeeds first time if no compile error
19511	Move "User Guide" higher in Help menu
19547	Fixed bug preventing users from finding	and - in the script editor
19564	Prevent error when deleting script editor tab
19585	Improve rendering of Interactive Tutorial on Windows when screenPixelScale > 1
19589	Fix 'put globalVar' in msg box
19627	Clear deleted objects from project browser correctly
19629	Check for changes in filename when getting object row in project browser
19749	Unsubscribe from ideExtensionLog when extension builder closes
19838	Add explicit 'select object' steps to the Interactive Tutorials to ensure the correct object is always selected.
19856	Prevent extension builder from duplicating logs
19888	Fix some minor typos and errors in interactive tutorials
19889	Allow tutorial instruction window to be moved
19940	Always use Return key in tutorial instructions instead of Enter key.
19942	Fix typos in BMI tutorial
19943	Fix ambiguity in a couple of BMI tutorial instructions
19967	Fix second parameter evaluation for script in message box
19978	Ensure default script is editable from within the Extensions Builder
19984	Ensure custom props value comparison is case-sensitive
19985	Remove errant group from the script editor
20039	Position tutorial controls in better location when using 'Do It For Me'
20040	Use smaller images of todo list on smaller screens
20041	Prevent tutorials breaking when stack name is changed

20044 Detect specific errors in user scripts in tutorial

20046 Ensure there are separate scripting / apply steps in tutorial

20071 Make default handler name text grey

20072 Add space above default handler list and before each name

20074 Prevent removal of initial P from default handler name

20077 Default to 3-column tools palette

20102 Don't shortcut 'is changed' property steps

20103 Clear highlights before epilogue of interactive tutorial

20112 Unlock cursor on tab-command-alt key

20117 Don't override existing users' backdrop setting

20133 Ensure cloning stacks or cards from the Project Browser works correctly

20170 Fixed incorrect name of PI template stack

20171 Make sure LiveCode 8+ launches correctly if only an old (livecode.rev) Preferences file is present

20330 Show error dialog when the name of the stack contains quotes

20345 Ensure the IDE reopens a DB connection if this was previously closed by the user

20431 Show ask and answer dialogs in correct location when the screen top is not 0

20535 Check for IDE stacks correctly when toggling `Suppress Messages`

20536 Check for IDE stacks correctly when suspending development tools

20647 Ensure Dictionary responds to cmd+W shortcut

20713 Enable lock/unlock text from popUp for fields

20894 [Start Center] Ensure creating new stack with tablet Landscape button works as expected

4010 Ensure Image/Object Library places images/objects only on user's stack

5787 Drag and drop stackfiles

6289 Ensure navigation with arrow keys works in the LiveCode Preferences window

LiveCode Builder changes

LiveCode Builder Standard Library

Foreign function interface

- When passing a Number to one of the foreign integer types (`LCInt`, `LCUInt`, `IntSize`, `UIntSize`), an error will be thrown if the value is outside the range of the requested type.

- The `IntSize` and `UIntSize` types can hold the full 64-bit integer range, however the maximum magnitude which is supported for converting to and from Number is 2^53. An error will be thrown for any conversions outside this range.

Specific LCB bug fixes (8.1.10-rc-1)

18856   Use cached numeric values when converting numbers to/from strings
19067   Ensure an error is thrown if there is no script access
19214   Increase usable range of IntSize and UIntSize types
19244   Nil pointers should bridge to nothing

LiveCode extension changes

Tree View widget

Limit display calculations

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

Specific extension bug fixes (8.1.10-rc-1)

16241   Runtime error when changing itemCount by more than one
18319   Prevent segmented control fill from bleeding outside border.
18391   Correctly order default marker styles
18500   Ensure color properties are documented correctly
18693   Prevent long delays when expanding arrays with many keys
19261   Clear selection when deleting selected node
19350   Remember iconPresetName in SVG Icon when reopening
19907   Prevent LCB error when deleting key at path with no selection
20534   Ensure segmented widget created in LC 9.x opens in LC 8.1.x too

Dictionary additions

- **deleteAudioclip** (*message*) has been added to the dictionary.
- **mobileSetFullScreenRectForOrientations** (*command*) has been added to the dictionary.

Previous release notes

- LiveCode 9.0.0 Release Notes
- LiveCode 8.1.9 Release Notes
- LiveCode 8.1.8 Release Notes
- LiveCode 8.1.7 Release Notes
- LiveCode 8.1.6 Release Notes
- LiveCode 8.1.5 Release Notes
- LiveCode 8.1.4 Release Notes
- LiveCode 8.1.3 Release Notes
- LiveCode 8.1.2 Release Notes
- LiveCode 8.1.1 Release Notes
- LiveCode 8.1.0 Release Notes
- LiveCode 8.0.2 Release Notes
- LiveCode 8.0.1 Release Notes
- LiveCode 8.0.0 Release Notes
- LiveCode 7.1.4 Release Notes
- LiveCode 7.1.3 Release Notes
- LiveCode 7.1.2 Release Notes
- LiveCode 7.1.1 Release Notes
- LiveCode 7.1.0 Release Notes
- LiveCode 7.0.6 Release Notes
- LiveCode 7.0.4 Release Notes
- LiveCode 7.0.3 Release Notes
- LiveCode 7.0.1 Release Notes
- LiveCode 7.0.0 Release Notes
- LiveCode 6.7.9 Release Notes
- LiveCode 6.7.8 Release Notes
- LiveCode 6.7.7 Release Notes
- LiveCode 6.7.6 Release Notes
- LiveCode 6.7.4 Release Notes
- LiveCode 6.7.2 Release Notes
- LiveCode 6.7.11 Release Notes
- LiveCode 6.7.10 Release Notes
- LiveCode 6.7.1 Release Notes
- LiveCode 6.7.0 Release Notes
- LiveCode 6.6.2 Release Notes
- LiveCode 6.6.1 Release Notes
- LiveCode 6.6.0 Release Notes
- LiveCode 6.5.2 Release Notes
- LiveCode 6.5.1 Release Notes
- LiveCode 6.5.0 Release Notes
- LiveCode 6.1.3 Release Notes
- LiveCode 6.1.2 Release Notes
- LiveCode 6.1.1 Release Notes
- LiveCode 6.1.0 Release Notes
- LiveCode 6.0.2 Release Notes
- LiveCode 6.0.1 Release Notes
- LiveCode 6.0.0 Release Notes