

LiveCode 6.5.0 Release Notes

Table of contents

Overview

Known issues

Platform support

- Windows

- Linux

- Mac

Setup

- Installation

- Uninstallation

Reporting installer issues

Activation

Multi-user and network install support (4.5.3)

Command-line installation

Command-line activation

Proposed changes

Engine changes

- Full screen scaling mode.

- Improved image editing tools.

- Take account of keyboard visibility in Android "effective working screenrect". Notify engine of changes to keyboard visibility.

- Crash when attempting to print to file on linux.

- Fullscreen modes cause clipped text on Windows

- Printing text to PDF on Windows can result in poor layout.

- Server graphics support

- The fullscreenModes are now camel-case.

- PCRE library updated to version 8.33

- libUrlSetSSLVerification now supported on mobile platforms

Resolution Independence

- New Graphics Layer

- Multiple Density Support

- Density Mapped Images

- Future Plans

 - More control over automatic scaling

 - Hi-DPI support on desktop platforms.

New global property colorDialogColors

Integration of revFont external

Enhanced 'filter' command

- Filtering items

- Matching regular expressions

- Storing the output in another container

- Adoption of 'convert' semantics

- Backward compatibility

Text Measurement

The optional *recursively* adverb has been added to union and intersect commands

Xpath functions

Syntax

Usage

Standalones now set default font settings the same as the IDE.

Setting the filename of an image which already has a filename causes the property to be unset and 'could not load image' in the result.

Setting the image filename to empty unsets the image text and vice-versa

Inconsistent rounding of floating point values.

Scrollbar properties not returned in correct format.

Specific bug fixes (6.5.0)

Specific bug fixes (6.5.0-rc-7)

Specific bug fixes (6.5.0-rc-6)

Specific bug fixes (6.5.0-rc-5)

Specific bug fixes (6.5.0-rc-4)

Specific bug fixes (6.5.0-rc-2)

Specific bug fixes (6.5.0-rc-1)

Specific bug fixes (6.5.0-dp-2)

Specific bug fixes (6.5.0)

Dictionary additions

Dictionary changes

Previous Release Notes

Overview

This document describes all the changes that have been made for LiveCode 6.5.0, including bug fixes and new syntax.

Known issues

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

Platform support

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

Windows

The engine supports the following Windows OSes:

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

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

Linux

The linux engine requires the following:

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

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

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

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

Mac

The Mac engine supports:

- 10.4.11 (Tiger) on Intel and PowerPC
- 10.5.8 and later (Leopard) on Intel and PowerPC
- 10.6.x (Snow Leopard) on Intel
- 10.7.x (Lion) on Intel
- 10.8.x (Mountain Lion) on Intel
- 10.9.x (Mavericks) on Intel

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

Setup

Installation

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

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

- Windows: <x86 program files folder>/RunRev/ LiveCode 6.5.0
- Linux: /opt/runrev/livecode-6.5.0
- Mac: /Applications/ LiveCode 6.5.0.app

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

- Windows: <user roaming app data folder>/RunRev/Components/LiveCode 6.5.0
- Linux: ~/.runrev/components/livecode-6.5.0
- Mac: ~/Applications/ LiveCode 6.5.0.app

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

Uninstallation

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

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

On Linux, the situation is currently less than ideal:

- open a terminal
- `cd` to the folder containing your rev install. e.g.

```
cd /opt/runrev/livecode-6.5.0
```

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

```
./ .setup.x86
```

- follow the on-screen instructions.

Reporting installer issues

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

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

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

- **Windows Vista/7:** <users folder>/<user>/AppData/Local/RunRev/Logs
- **Linux:** <home>/runrev/logs
- **Mac:** <home>/Library/Application Support/Logs/RunRev

Activation

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

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

Multi-user and network install support (4.5.3)

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

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

Command-line installation

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

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

```
<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 <i>location</i>	The location to install into. If not specified, the location defaults to those described in the <i>Layout</i> section above.
-log <i>logfile</i>	A file to place a log of all actions in. If not specified, no log is generated.

Note that the command-line variant of the installer does not do any authentication. Thus, if you wish to install to an admin-only location you will need to be running as administrator before executing the command. As the installer is actually a GUI application, it needs to be run slightly differently from other command-line programs.

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

On Windows, you need to do:

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

On Mac, you need to do:

```
"<installerexe>/Contents/MacOS/installer" install noui options
```

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

Command-line activation

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

On both platforms, the command is of the form:

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

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

This action can be undone using the following command:

```
<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.

Proposed changes

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

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

Engine changes

Full screen scaling mode. (6.5.0-rc-6)

IMPORTANT: The *showAll* mode has been renamed for this release to the more descriptive *letterbox* mode. We plan to add a revised version of the *showAll* mode in a future release that does not draw the black bars that appear in the current implementation but instead displays any objects that lie outside the defined rect of the stack.

To limit the impact of this change we have decided to rename *showAll* without retaining it as a synonym of *letterbox* to avoid changing the behaviour of that mode after the release of LiveCode 6.5.0-gm-1.

If you are using the "showAll" mode in your stacks, please update them to use the new syntax.

There are multiple ways in which a stack can be resized or scaled to take full advantage of the available screen space. The full screen scaling mode allows the developer to choose the most appropriate for their application.

The fullscreen scaling mode of a stack can be changed by setting its **fullscreenmode** property to one of the following values:

- empty (default) - the existing behaviour - the stack is resized (not scaled) to fit the screen.
- "exactFit" - scale the stack to fill the screen. This will stretch the stack if the aspect ratio of the screen does not match that of the stack.
- "letterbox" - scale the stack preserving aspect ratio so all content is visible. Black bars will fill any empty space if the screen & stack aspect ratios do not match.
- "noBorder" - scale the stack to fill the screen preserving aspect ratio. If the stack & screen aspect ratios do not match, the left / right or top / bottom extremes of the stack will not be

visible.

- "noScale" - the stack will not be scaled, being centered on the screen instead.

Note that the **fullscreenmode** only takes effect when a stack is fullscreen. This will be the case on mobile platforms where stacks are always fullscreen, or on the desktop when the **fullscreen** property of the stack is set to true.

The fullscreen scaling mode is available on all desktop and mobile platforms and operates independently from Hi-DPI support.

Improved image editing tools. (6.5.0-rc-5)

When using an image editing tool involving a brush (the brush / the spray / the eraser), the alpha channel will now be properly utilized and alpha blending will occur.

The image editing tools now work on mobile platforms.

Take account of keyboard visibility in Android "effective working screenrect". Notify engine of changes to keyboard visibility. (6.5.0-rc-5)

Fixes issue of stack rect being incorrect if orientation change occurs while the keyboard is visible

Crash when attempting to print to file on linux. (6.5.0-rc-4)

A crash can occur when printing on linux if the engine is unable to create the file for printed output. This typically occurs if the defaultFolder has not changed to somewhere writable by the time 'print' is invoked.

Fullscreen modes cause clipped text on Windows (6.5.0-rc-2)

In order for fullscreen scaling to work on Windows it is necessary to use device-independent text metrics - the same as when using formatForPrinting. Therefore, on Windows, when you set the fullscreenmode to something other than resize, the stack will change to using device-independent metrics for all text (similar to when setting formatForPrinting).

Printing text to PDF on Windows can result in poor layout. (6.5.0-rc-2)

You should now set 'formatForPrinting' to true for stacks you are printing to PDF on Windows. This will ensure the metrics of the text match those which PDFs expect. Note that 'formatForPrinting' on Windows will now display on screen with the same metrics.

Server graphics support (6.5.0-rc-1)

Support for graphic rendering has been to the server engines. This allows users to use the export snapshot command as they would on the desktop and mobile platforms. For example, the following command can be called from within a server script in order to create a PNG image of the first card of the given stack (note that by default, server stacks have a black background).

```
export snapshot from card 1 of stack "graphics.livecode" to file
"graphics.png" as PNG
```

As part of these changes, the Linux server engine now requires PangoFT2 and Glib for text rendering.

The fullscreenModes are now camel-case. (6.5.0-rc-1)

For consistency and to ease usage in non-variable checking mode, then fullscreenModes have been changed to be exactFit, showAll, noBorder and noScale. The previous (two word) variants will be removed for release.

PCRE library updated to version 8.33 (6.5.0-dp-2)

The version of the PCRE (Perl Compatible Regular Expressions) library the engine uses has been updated to 8.33. This affects the functionality of matchText, replaceText and 'filter with regex pattern'. The previous version used was version 6.7 and full details of the changes since then can be found at <http://www.pcre.org/news.txt> and <http://www.pcre.org/changelog.txt>

libUrlSetSSLVerification now supported on mobile platforms (6.5.0-dp-2)

You can now use libUrlSetSSLVerification on Android and iOS to enable or disable SSL verification checking. Turning verification off turns off both checking of server-side certificates and hostname matching.

Resolution Independence (6.5.0-dp-2)

The primary feature included in LiveCode 6.5 is support for resolution independence. Resolution independence means that an app can be designed using an abstract pixel density and then have that automatically map to the pixel density of the display with no loss of quality.

For example, iPhones have retina displays which are twice the density of previous generations. This greater density isn't intended to provide more screen real-estate, but instead provide a crisper display. An app should be designed at the standard density (about 163dpi) leaving it to the OS to scale the rendering to enable the app to benefit from the greater density.

New Graphics Layer

The first step to supporting resolution independence was to completely refactor LiveCode's graphics layer. This involved writing and integrating an entirely new 2D graphics library that allows for scaled drawing. In addition to 2D graphic rendering, the library also handles text and image rendering. As such, nearly all aspects of LiveCode's drawing routines have been touched.

Since all of the updates are internal, the end LiveCode developer should see no major changes: Where possible, we've tried to match previous behaviors as closely as possible. However, in the early DPs, we do expect some rendering irregularities.

Though the primary reason for the updates is to support resolution independence, we do get the side benefits of having a modern 2D graphics library. These include a clean developer API allowing for easy integration into other modules, potential performance improvements and support for graphic rendering on the server platforms.

Multiple Density Support

Most OSs support multiple pixel densities, with there being a "natural density" and a notion of Hi-DPI. DP1 of LiveCode 6.5 has support for multiple densities on Android and iOS.

For iOS, the screen is either retina or non-retina. The natural density is the non-retina resolutions. Retina screens are considered to be 2x the density of non-retina.

For Android, screens usually fall in to one of four density categories - low (0.75x), medium (1x), high (1.5x), extra-high (2x). Medium is considered to be the natural density. Some devices have a "TV" screen density, at 1.33x scale.

When coding for multiple densities, we take the notion of a "pixel" from a stack's point of view to be "a pixel at the natural density of the OS", what we refer to as a point. A scale factor is then applied on output to map to that of the screen density of the device.

This is an entirely invisible process. From the point of view of the app everything appears as if the screen was the "natural" density. In particular:

- import / export snapshot work in points, not pixels - i.e they produce images at point resolution
- imageData (maskData/alphaData) of images work in points, not pixels
- intersections work in points
- hit detection works in points
- the screenRect properties and stack rect are given in points

Android screen sizes prior to the introduction of resolution independence were given in pixels, rather than points. As a result of this many Android devices with high density displays will now report an apparently smaller screen size. This may require modifications to existing stacks that were created with the larger pixel size in mind.

Density Mapped Images

Whilst text and vector operations scale naturally, this is not true of images. To take full advantage of Hi-DPI,

images must be provided at appropriate sizes for different densities. The developer can do this by having multiple image files in the same location named appropriately (with the image object referencing the image file at the natural density). The naming convention is as follows:

- `<image>@ultra-low.<ext>` - 0.25x
- `<image>@extra-low.<ext>` - 0.5x
- `<image>@low.<ext>` - 0.75x
- `<image>@medium.<ext>` / `<image>.<ext>` - 1x
- `<image>@high.<ext>` - 1.5x
- `<image>@extra-high.<ext>` / `<image>@2x` - 2x
- `<image>@ultra-high.<ext>` - 4x

When an image is required, the current scale factor is rounded up to the nearest standard density (one of 0.25, 0.5, 0.75, 1, 1.5, 2 and 4). The image with the lowest scale factor that is greater or equal to the nearest standard density is then selected. For example, if the scale factor is 1.75 and there is an "extra-high" image available that is used.

Images should be created at appropriately scaled pixel sizes - if the natural size of an image is x points wide and y points high, the scaled versions should be $x \text{ scale pixels wide and } y \text{ scale pixels high}$ in order to have the same natural size. Developers should take care that the natural size of the image scales nicely to integer pixel sizes, as the logical point size of the image will be rounded up to the nearest integer. If this is not practical for a given image, the developer may need to design their apps to handle slightly different resulting image sizes.

A useful rule of thumb is for images to have natural width and height values that are multiples of four, this way all the supported densities will result in integer pixel sizes for each version of the image, and the resulting image will have the same logical point size as all display densities.

Future Plans

More control over automatic scaling

Currently, on Android and iOS the scale factor is automatically applied. On iOS, this can be overridden by calling the existing command "iphoneUseDeviceResolution true", which will turn off scaling so one point is equal to one pixel. This capability will be generalized to all platforms supporting resolution independence, and extended to allow configuration of the display scale.

Hi-DPI support on desktop platforms.

Automatically scale stacks on desktop systems with high resolution displays. This will function in the same way as the current support for mobile devices.

Support for automatic scaling will be added for desktop operating systems that support high resolution displays. This currently includes Windows 7 & 8 and OSX Mountain Lion.

New global property `colorDialogColors` (6.5.0)

The `colorDialogColors` provides a way to get and set the custom colors a user defined in the color dialog or provide a default set of colors for the user to pick from. The property currently only has an effect on Windows.

Integration of `revFont` external (6.5.0)

To load a font into memory use:

start using font file *fontFile* [globally]

To unload a font from memory use:

stop using font file *fontFile*

To determine the currently loaded font files use:

the `fontFilesInUse` property

Enhanced 'filter' command (6.5.0)

The **filter** command was enhanced to support:

- filtering items in addition to lines
- matching a regular expression in addition to wildcard patterns
- storing the output in another container using an optional 'into' clause
- as well as the adoption of 'convert' semantics.

The new syntax is:

```
filter [ { lines | items } of ] <source_container_or_expr> { with | without |
[ not ] matching } [ { wildcard | regex } [ pattern ] ] <pattern> [ into
<target_container> ]
```

Note that the implementation is backward compatible as:

- the default chunk type is 'lines'
- and the default pattern type is 'wildcard'.

Filtering items

In previous versions, the 'filter' command only supported the filtering of lines. Now you can also filter items in a source.

For example, for a variable 'theList' containing a comma-separated list of strings:

```
foo,bar,baz,qux,quux,corge,grault,garp,ly,waldo,fred,plugh,xyzyzy,thud
```

The script line:

```
filter items of theList with "b
```

Would result in the variable 'theList' containing:

```
bar,baz
```

Matching regular expressions

In previous versions, the 'filter' command only supported matching a 'wildcard' pattern. Now you can use regular expression pattern matching as well.

For example, for a variable 'theList' containing a comma-separated list of strings:

```
foo,bar,baz,qux,quux,corge,grault,garp,ly,waldo,fred,plugh,xyzyzy,thud
```

The script line:

```
filter items of theList with regex pattern "b.
```

Would result in the variable 'theList' containing:

```
bar,baz
```

Note that the keyword 'pattern' is optional syntactic sugar to clarify the intent of the script.

Storing the output in another container

In previous versions, the 'filter' command always replaced the contents of the original container. Now you can opt to store the output in a separate container, allowing you to easily retain both the original and filtered data in separate variables.

For example, for a variable 'theList' containing a comma-separated list of strings:

```
foo,bar,baz,qux,quux,corge,grault,garply,waldo,fred,plugh,xyzy,thud
```

The script line:

```
filter items of theList with "b
```

Would result in the variable 'theFilteredList' containing:

```
bar,baz
```

While the variable 'theList' still contains the original unfiltered data.

Adoption of 'convert' semantics

In previous versions, the 'filter' command only supported containers as input, not expressions. Now you can use any expression as input, and the output is stored in a separate container (if the 'into' clause is used) or in the special 'it' variable.

For example, for a variable 'theFirstList' containing a comma-separated list of strings:

```
foo,bar,baz,qux,quux,corge,grault
```

And a second variable 'theSecondList' containing another comma-separated list of strings:

```
garply,waldo,fred,plugh,xyzy,thud
```

The script line:

```
filter items of theFirstList & comma & theSecondList with "b
```

Would result in the variable 'theFilteredList' containing:

```
bar,baz
```

On the other hand, the script line:

```
filter items of theFirstList & comma & theSecondList with "b
```

Would result in the variable 'it' containing:

```
bar,baz
```

Backward compatibility

As stated above, the implementation is backward compatible.

This means that the script line:

```
filter theList with "b"
```

Is equivalent to the following, more explicit variants:

```
filter lines of theList with wildcard "b"
```

```
filter lines of theList matching wildcard pattern "b"
```

Likewise, the script line:

```
filter theList without "b"
```

Is equivalent to the following, more explicit variants:

```
filter lines of theList without wildcard "b"
```

```
filter lines of theList not matching wildcard pattern "b"
```

Text Measurement (6.5.0)

There are two new functions **measureText()** and **measureUnicodeText()** which can be used to measure the area required to draw the text using the effective font attributes of a given object.

measureText(text,object reference,[mode])

measureUnicodeText(unicodeText,object reference,[mode])

The mode may be one of:

- width - the functions return the width of the text.
- size - the functions return the width,height of the text.
- bounds - the functions return a rectangle of 0, - ascent, width, descent.

If no mode is specified the functions default to *width* mode. Ascent and descent are relative to a 0 baseline that the text would be drawn on.

The optional recursively adverb has been added to union and intersect commands (6.5.0)

Using the *recursively* adverb causes the union or intersect command to recurse through the arrays rather than only working on the root elements.

Xpath functions (6.5.0)

The following xpath functions were added to the engine:

- 'revXMLEvaluateXPath'
- 'revXMLDataFromXPathQuery'

In addition, several synonyms for existing commands are now available for consistency:

- 'revXMLCreateTree'
- 'revXMLCreateTreeWithNamespaces'
- 'revXMLCreateTreeFromFile'
- 'revXMLCreateTreeFromFileWithNamespaces'
- 'revXMLDeleteTree'

- 'revXMLAppend'
- 'revXMLDeleteAllTrees'
- 'revXMLAddNode'
- 'revXMLDeleteNode'
- 'revXMLInsertNode'
- 'revXMLMoveNode'
- 'revXMLCopyNode'
- 'revXMLCopyRemoteNode'
- 'revXMLMoveRemoteNode'
- 'revXMLPutIntoNode'
- 'revXMLSetAttribute'

Syntax

revXMLEvaluateXPath(pDocID, pXPathExpression [, delimiter])
 revXMLDataFromXPathQuery(pDocID, pXPathExpression [, delimiter])

Usage

Given tXMLData as

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
<book category="COOKING">
<title lang="en">Everyday Italian</title>
<author>Giada De Laurentiis</author>
<year>2005</year>
<price>30.00</price>
</book>
<book category="CHILDREN">
<title lang="en">Harry Potter</title>
<author>J K. Rowling</author>
<year>2005</year>
<price>29.99</price>
</book>
<book category="WEB">
<title lang="en">XQuery Kick Start</title>
<author>James McGovern</author>
<author>Per Bothner</author>
<author>Kurt Cagle</author>
<author>James Linn</author>
<author>Vaidyanathan Nagarajan</author>
<year>2003</year>
<price>49.99</price>
</book>
<book category="WEB">
<title lang="en">Learning XML</title>
<author>Erik T. Ray</author>
<year>2003</year>
<price>39.95</price>
</book>
</bookstore>
```

Then:

```
put revXMLCreateTree(tXMLData,false,true,false) into pDocID
put "/bookstore/book[price<50]" into pXPathExpression
put revEvaluateXPath(pDocID, pXPathExpression)
```

Gives you:

```
/bookstore/book[1]
/bookstore/book[2]
/bookstore/book[3]
/bookstore/book[4]
```

And:

```
put "/bookstore/book[price<30]/title" into pXPathExpression
put revDataFromXPathQuery(pDocID, pXPathExpression)
```

Gives you 'Harry Potter'

Standalones now set default font settings the same as the IDE. (6.5.0)

The engine now configures default font settings inline with the IDE as follows:

- Windows Vista and later - 12pt Segoe UI
- Windows pre-Vista - 11pt Tahoma
- Mac OS X - 11pt Lucida Grande
- Linux - 12pt Helvetica
- Other platforms - 14pt Helvetica

Setting the filename of an image which already has a filename causes the property to be unset and 'could not load image' in the result. (6.5.0)

Previously setting the filename of image which had a non-empty filename property would cause the property to be unset and an error in the result. Since the purpose of setting the filename to empty is to clear the reference, this behavior has been changed so no error is generated (in the result) in this case.

Setting the image filename to empty unsets the image text and vice-versa (6.5.0)

Previously, setting the filename of an image where it's filename was already empty would cause the text to be unset and vice-versa. This goes against the idea that setting a property to its existing value should have no effect, and is inconsistent with similar property pairs such as `foreColor/forePattern`.

Now, if you attempt to set the filename to empty when it is already empty, the action will have no effect. Similarly if you attempt to set the text to empty when it is already empty, it will have no effect.

Inconsistent rounding of floating point values. (6.5.0)

Previously when setting properties expecting integers, some properties would truncate real values, others would round-to-nearest. The rounding mode for all conversions is now consistent - round-to-nearest.

Scrollbar properties not returned in correct format. (6.5.0)

Previously these properties returned different values when obtained as numbers and string - as strings they

returned the value rounded. Now they return the same value - rounded to a precision specified by the numberFormat of the scrollbar.

Specific bug fixes (6.5.0)

(bug fixes specific to the current build are highlighted in bold, reverted bug fixes are stricken through)

- 11523** **3D borders render incorrectly**
- 11522** **Crash when getting the filename of an image which references a file that does not exist**
- 11512** **1 pixel borders don't scale correctly in fullscreen mode**
- 11511** **Fix a crash when retrieving contacts imported from Outlook into the iOS address book**
- 11501** **Certain non-antialiased polygons with 1 pixel borders fill incorrectly**
- 11496** **Redrawing gradients when accelerated rendering is on can cause lines to appear in certain cases**
- 11485** **Mobile native controls are positioned incorrectly in fullscreen mode**
- 11458** **Text size element of styled text array is not converted to numeric format**

Specific bug fixes (6.5.0-rc-7)

- 11494 Crash when calling intersects on a button with an icon.
- 11486 Angled images not rendering correctly in certain situations.
- 11475 Field htmltext has invalid color attribure values
- 11428 Fix crash on Android calling mobilePlaySoundOnChannel with invalid path
- 11395 Grapic markers are drawn incorrectly
- 11278 Android mobilePick can only be called once
- 11149 revDatabaseConnectResult always returns empty

Specific bug fixes (6.5.0-rc-6)

- 11476 Freeze when drawing certain gradients
- 11472 Linux "Open Stack..." dialog does not show "All Files" option.

Specific bug fixes (6.5.0-rc-5)

- 11471 Loss of quality for certain gradient types
- 11470 Regular polygons with a line size greater than 1 clip
- 11469 Ovals are filled incorrectly when using startAngle/arcAngle
- 11464 Fix crash when printing an RLE-encoded image on OSX
- 11457 Antialised single point lines draw across pixels
- 11456 OS X tab buttons clip for certain text sizes
- 11445 Fix image scaling with "best" resizequality rounding scale factors down to integer values.
- 11444 Odd letter spacing in text when rendered in certain fonts at certain sizes.
- 10485 Take account of keyboard visibility in Android "effective working screenrect". Notify engine of changes to keyboard visibility.

Specific bug fixes (6.5.0-rc-4)

- 11436 Standalone builder still has option for Armv6 iOS builds
- 11426 When drawing opaque round rectangles the fill overlaps the stroke
- 11422 Multiply blend mode incorrect
- 11415 Random issues with text layout and alignment.

- 11413 Text clipped vertically on iOS for certain fonts (updated fix for iOS 6)
- 11320 Player rect not adjusted by stack scroll when redrawing on OSX
- 11197 Crash when attempting to print to file on linux.

Specific bug fixes (6.5.0-rc-2)

- 11397 Imagedata of resized images returned at wrong scale
- 11393 Fullscreen modes cause clipped text on Windows
- 11390 Fix incorrect cropping of resized image
- 11389 Dashed borders on oval graphics do not start at the correct location
- 11388 1 pixel non-antialiased lines draw incorrectly in certain circumstances
- 11360 Linux theming does not correctly work on non 32bit displays
- 11355 Focus border on Windows default button has artefacts.
- 10508 Printing text to PDF on Windows can result in poor layout.

Specific bug fixes (6.5.0-rc-1)

- 11367 Gradient inspector does not draw fully
- 11359 Redraw issues with inner glow and inner shadow.
- 11351 "recursively" option for array union and intersect not working.
- 11349 Content outside of card rect can be seen in show all fullscreenmode when acceleratedRendering is true.
- 11338 Character backColor not rendered in fields.
- 11336 The fullscreenModes are now camel-case.
- 11335 In fullscreen mode on Mac
- 11332 Text not rendered in correct location in -ui mode when taking snapshot of object.
- 11330 Visual effects not displayed correctly on iOS when fullscreenmode is 'no border'
- 11329 Dynamic layers do not render correctly when fullscreenmode causes stack to not be at top-left of screen.
- 11327 Opening large scripts takes longer in 6.5 than 6.1.2
- 11325 Graphic effects rendered incorrectly when non-Normal blend mode used.
- 11324 Borders still rendered even if borderWidth == 0.
- 11315 Instability of LiveCode when gradients are being / have been displayed.
- 11314 Image transparency is lost when setting the imagedata
- 11299 Answer folder for XCode app bundle has a typo
- 11198 Fix color issues on PowerPC Mac
- 11174 Conical spiral XY and sqrtXY gradients render incorrectly.

Specific bug fixes (6.5.0-dp-2)

- 11310 Italic fonts clipped on right hand side on iOS
- 11309 Text clips for certain fonts on Mac
- 11297 Setting the color of a graphic effect resets its opacity
- 11283 Italic fonts are not rendered on Mac.
- 11280 Project browser does not expand cards of substacks
- 11241 Non-3D borders are always black.
- 11227 Android native controls are positioned incorrectly.
- 11221 Field background pattern not working.

- 11179 Images with an angle other than 0 are not drawn.
- 11178 Import from paint always fails.
- 11176 MouseDoubleUp messages are not sent on Linux.

Specific bug fixes (6.5.0)

(bug fixes specific to the current build are highlighted in bold, reverted bug fixes are stricken through)

- 10995 **Standalones now set default font settings the same as the IDE.**
- 10980 **Setting the filename of an image which already has a filename causes the property to be unset and 'could not load image' in the result.**
- 10977 **Setting the image filename to empty unsets the image text and vice-versa**
- 10964 **Inconsistent rounding of floating point values.**
- 6720 **Scrollbar properties not returned in correct format.**

Dictionary additions

- **resolve image** (*command*) has been added to the dictionary.
- **start using font** (*command*) has been added to the dictionary.
- **stop using font** (*command*) has been added to the dictionary.
- **measureText** (*function*) has been added to the dictionary.
- **measureUnicodeText** (*function*) has been added to the dictionary.
- **revXMLDataFromXPathQuery** (*function*) has been added to the dictionary.
- **revXMLEvaluateXPath** (*function*) has been added to the dictionary.
- **xsltApplyStylesheet** (*function*) has been added to the dictionary.
- **xsltApplyStylesheetFromFile** (*function*) has been added to the dictionary.
- **xsltLoadStylesheet** (*function*) has been added to the dictionary.
- **xsltLoadStylesheetFromFile** (*function*) has been added to the dictionary.
- **colorDialogColors** (*property*) has been added to the dictionary.
- **fontFilesInUse** (*property*) has been added to the dictionary.
- **fullscreenmode** (*property*) has been added to the dictionary.

Dictionary changes

- The entry for **edit** (*command*) has been updated.
- The entry for **filter** (*command*) has been updated.
- The entry for **intersect** (*command*) has been updated.
- The entry for **libURLSetSSLVerification** (*command*) has been updated.
- The entry for **revXMLAddDTD** (*command*) has been updated.
- The entry for **revXMLAddNode** (*command*) has been updated.
- The entry for **revXMLAppend** (*command*) has been updated.
- The entry for **revXMLCopyNode** (*command*) has been updated.
- The entry for **revXMLDeleteAllTrees** (*command*) has been updated.
- The entry for **revXMLDeleteNode** (*command*) has been updated.
- The entry for **revXMLDeleteTree** (*command*) has been updated.
- The entry for **revXMLInsertNode** (*command*) has been updated.
- The entry for **revXMLMoveNode** (*command*) has been updated.
- The entry for **revXMLPutIntoNode** (*command*) has been updated.
- The entry for **revXMLSetAttribute** (*command*) has been updated.

- The entry for **union** (*command*) has been updated.
- The entry for **revXMLAttribute** (*function*) has been updated.
- The entry for **revXMLAttributeValues** (*function*) has been updated.
- The entry for **revXMLAttributes** (*function*) has been updated.
- The entry for **revXMLChildContents** (*function*) has been updated.
- The entry for **revXMLChildNames** (*function*) has been updated.
- The entry for **revXMLCreateTree** (*function*) has been updated.
- The entry for **revXMLCreateTreeFromFile** (*function*) has been updated.
- The entry for **revXMLCreateTreeFromFileWithNamespaces** (*function*) has been updated.
- The entry for **revXMLCreateTreeWithNamespaces** (*function*) has been updated.
- The entry for **revXMLFirstChild** (*function*) has been updated.
- The entry for **revXMLMatchingNode** (*function*) has been updated.
- The entry for **revXMLNextSibling** (*function*) has been updated.
- The entry for **revXMLNodeContents** (*function*) has been updated.
- The entry for **revXMLNumberOfChildren** (*function*) has been updated.
- The entry for **revXMLParent** (*function*) has been updated.
- The entry for **revXMLPreviousSibling** (*function*) has been updated.
- The entry for **revXMLRootNode** (*function*) has been updated.
- The entry for **revXMLText** (*function*) has been updated.
- The entry for **revXMLTree** (*function*) has been updated.
- The entry for **revXMLTrees** (*function*) has been updated.
- The entry for **revXMLValidateDTD** (*function*) has been updated.
- The entry for **editScript** (*message*) has been updated.
- The entry for **revEndXMLNode** (*message*) has been updated.
- The entry for **revStartXMLData** (*message*) has been updated.
- The entry for **revStartXMLNode** (*message*) has been updated.
- The entry for **revXMLEndTree** (*message*) has been updated.
- The entry for **revXMLStartTree** (*message*) has been updated.
- The entry for **filename** (*property*) has been updated.
- The entry for **secureMode** (*property*) has been updated.
- The entry for **securityPermissions** (*property*) has been updated.

Previous Release Notes

6.5.0 Release Notes	http://downloads.livecode.com/livecode/6_5_0/LiveCodeNotes-6_5_0.pdf
6.1.3 Release Notes	http://downloads.livecode.com/livecode/6_1_3/LiveCodeNotes-6_1_3.pdf
6.1.2 Release Notes	http://downloads.livecode.com/livecode/6_1_2/LiveCodeNotes-6_1_2.pdf
6.1.1 Release Notes	http://downloads.livecode.com/livecode/6_1_1/LiveCodeNotes-6_1_1.pdf
6.1.0 Release Notes	http://downloads.livecode.com/livecode/6_1_0/LiveCodeNotes-6_1_0.pdf
6.0.2 Release Notes	http://downloads.livecode.com/livecode/6_0_2/LiveCodeNotes-6_0_2.pdf
6.0.1 Release Notes	http://downloads.livecode.com/livecode/6_0_1/LiveCodeNotes-6_0_1.pdf
6.0.0 Release Notes	http://downloads.livecode.com/livecode/6_0_0/LiveCodeNotes-6_0_0.pdf