

LiveCode 6.5.0-dp-1 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

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

Resolution Independence

New Graphics Layer

Multiple Density Support

Density Mapped Images

Future Plans

More control over automatic scaling

Full screen scaling mode.

Hi-DPI support on desktop platforms.

Server graphics.

Xpath functions

Syntax

Usage

Resolving object chunks does not throw an error for long chunk references.

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

Getting 'the short name' of an object returns the abbrev id if the name property is empty

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.

Dictionary additions

Dictionary changes

Previous Release Notes

Overview

This document describes all the changes that have been made for LiveCode 6.5.0-dp-1, 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 2000 SP4
- 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

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

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-dp-1
- Linux: /opt/runrev/livecode-6.5.0-dp-1
- Mac: /Applications/ LiveCode 6.5.0-dp-1.app

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

- Windows: <user roaming app data folder>/RunRev/Components/LiveCode 6.5.0-dp-1
- Linux: ~/.runrev/components/livecode-6.5.0-dp-1
- Mac: ~/Applications/ LiveCode 6.5.0-dp-1.app

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

Uninstallation

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

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

On Linux, the situation is currently less than ideal:

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

```
cd /opt/runrev/livecode-6.5.0-dp-1
```

- 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

New global property `colorDialogColors` (6.5.0-dp-1)

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-dp-1)

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-dp-1)

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,xyzz,y,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,garply,waldo,fred,plugh,xyzzzy,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,xyzzzy,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,xyzzzy,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-dp-1)

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-dp-1)

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

Resolution Independence (6.5.0-dp-1)

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 (planned for a future release).

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 *scale pixels wide* and y *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.

Full screen scaling mode.

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 will allow the developer to choose the most appropriate for their application:

- empty (default) - the existing behaviour - the stack is resized (not scaled) to fit the screen.
- "exact fit" - 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.
- "show all" - scale the stack preserving aspect ratio so all content is visible. Some blank space may remain if the screen & stack aspect ratios do not match.
- "no border" - 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.

- "no scale" - the stack will not be scaled, being centered on the screen instead.

This will be available on all desktop platforms and operates independently from Hi-DPI support.

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.

Server graphics.

Support for creating images for output on a web server, for example by composing a set of objects and using "export snapshot ..." to generate the image data.

Xpath functions (6.5.0-dp-1)

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'

Resolving object chunks does not throw an error for long chunk references. (6.5.0-dp-1)

When using named object chunks such as 'control <name>' or 'field <name>', the engine only accepts <name> of the form:

- The short name of the object (e.g. 'foo')
- The name of the object (e.g. 'field "foo"')

Previously, in the case of matching a name, the engine would only match the prefix ignoring everything after it.

This resulted in things like 'field the long id of field "foo"' potentially resolving incorrectly.

This issue has now been fixed - when using named object chunks, <name>, must either be just a name, or a string of the form '<type> "<name>"'.

To resolve long id/long name references, there is no need for a chunk and you should just use the variable directly:

```
put the long id of field "foo" into tFooRef
answer the short name of tFooRef
```

Standalones now set default font settings the same as the IDE. (6.5.0-dp-1)

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

Getting 'the short name' of an object returns the abbrev id if the name property is empty (6.5.0-dp-1)

Previously the empty name for objects was not allowed, and the abbreviated id was returned instead for both 'the name' and 'the short name' of an object. This change allows objects to have the empty name, and adjusts the properties accordingly: 'the name' of such an object returns '<object type> ""', and 'the short name' returns empty.

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-dp-1)

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-dp-1)

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-dp-1)

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-dp-1)

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.

Dictionary additions

- **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.

Dictionary changes

- The entry for **filter** (*command*) has been updated.
- The entry for **intersect** (*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 **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.

Previous Release Notes

- 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