LiveCode 7.0.6-rc-3 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
- Engine changes
  - 'sort cards of background ...' crashes
  - Initializers for constants and locals too strict
  - Cannot specify text encoding of server script file.
  - ! /usr/bin/lcserver
  - -*- coding: utf-8 -*-
  - BidiDirection
  - Enable "umask" property on OS X
  - File format change
  - Array element pass by reference
  - Hebrew text is shown in reverse character order on Android
  - Unicode Support
  - Unicode and LiveCode
  - Creating Unicode Apps
    - New & Existing apps - things to look out for
  - New Commands, Functions & Syntax
    - Chunk expressions: byte, char, codepoint, codeunit
    - Chunk expressions: paragraph, sentence and trueWord
    - Synonym: segment
    - Property: the formSensitive
    - Command: open file/process/socket ... for <encoding> text
    - Functions: textEncode, textDecode
    - Functions: numToCodepoint, codepointToNum
    - Functions: numToNativeChar, nativeCharToNum
    - Function: normalizeText
    - Function: codepointProperty
  - Updated Functions
  - Function: binaryEncode
  - Function: binaryDecode
  - Deprecated Features

Functions: numToChar, charToNum
Property: useUnicode
Functions: uniEncode, uniDecode
Function: measureUnicodeText
Properties: unicodeText, unicodeLabel, unicodeTitle, unicodeTooltip, unicodePlainText, unicodeFormattedText
Specific bug fixes (7.0.6-rc-3)
Specific bug fixes (7.0.6-rc-2)
Specific bug fixes (7.0.6-rc-1)
Specific bug fixes (7.0.5-rc-3)
Specific bug fixes (7.0.5-rc-1)
Specific bug fixes (7.0.4)
Specific bug fixes (7.0.4-rc-3)
Specific bug fixes (7.0.4-rc-2)
Specific bug fixes (7.0.4-rc-1)
Specific bug fixes (7.0.2-rc-2)
Specific bug fixes (7.0.2-rc-1)
Specific bug fixes (7.0.1)
Specific bug fixes (7.0.1-rc-4)
Specific bug fixes (7.0.1-rc-3)
Specific bug fixes (7.0.1-rc-2)
Specific bug fixes (7.0.1-rc-1)
Specific bug fixes (7.0.0)
Specific bug fixes (7.0.0-rc-3)
Specific bug fixes (7.0.0-rc-2)
Specific bug fixes (7.0.0-dp-10)
Specific bug fixes (7.0.0-dp-9)
Specific bug fixes (7.0.0-dp-8)
Specific bug fixes (7.0.0-dp-7)
Specific bug fixes (7.0.0-dp-6)
Specific bug fixes (7.0.0-dp-5)
Specific bug fixes (7.0.0-dp-4)
Specific bug fixes (7.0.0-dp-3)
Specific bug fixes (7.0.0-dp-2)
IDE changes
Specific bug fixes (7.0.4-rc-1)
LiveCode Builder changes
Dictionary changes
Previous Release Notes

Overview

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

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

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

Known issues

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

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

Platform support

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

Windows

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:

- Supported architectures:

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

- Common requirements for GUI functionality:

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

- Requirements for 32-bit Intel/AMD:

glibc 2.3.6 or later

- *Requirements for 64-bit Intel/AMD:*

glibc 2.15 or later

- Requirements for ARMv6:

glibc 2.7 or later
Note: The GUI requirements are also required by Firefox and Chrome, so if your Linux distritution runes 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: It may be possible to compile and run LiveCode Community 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*

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 7.0.6-rc-3
- Linux: /opt/runrev/livecode-7.0.6-rc-3
- Mac: /Applications/ LiveCode 7.0.6-rc-3.app

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

- Windows: <user roaming app data folder>/RunRev/Components/LiveCode 7.0.6-rc-3
- Linux: ~/.runrev/components/livecode-7.0.6-rc-3
- Mac: ~/Applications/ LiveCode 7.0.6-rc-3.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-7.0.6-rc-3
```

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

Note that the command-line variant of the installer does not do any authentication. Thus, if you wish to install to an admin-only location you will need to be running as administrator before executing the command.
As the installer is actually a GUI application, it needs to be run slightly differently from other command-line programs.
In what follows <installerexe> should be replaced with the path of the installer executable or app (inside the DMG) that has been downloaded.
On Windows, you need to do:
start /wait <installerexe> install noui *options*
On Mac, you need to do:
“<installerexe>/Contents/MacOS/installer” install noui *options*

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

Command-line activation

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

Engine changes

'sort cards of background ...' crashes (7.0.4-rc-3)

You cannot 'sort cards of background' so this now throws an error, rather than crashing.

Initializers for constants and locals too strict (7.0.1-rc-2)

Initializers used for constants and locals can now be of one of the following forms:

- number
- +number
- -number
- quoted literal
- if explicitVars is false then an unquoted literal
- if explicitVars is true then any unquoted literal that would evaluate to the same string

The last rule means that when explicitVars is true you can use constants such as true and false without quotes.

Cannot specify text encoding of server script file. (7.0.1-rc-1)

You can now specify the text encoding of a server script file - that is a file that begins with #! and is passed to LiveCode Server.

To specify the text encoding of a server script file, the second line of the file must be a comment and contain the string coding=<encoding> or coding:<encoding>.

For example, to specify a script file as being encoded by UTF-8 one could use:

! /usr/bin/lcserver

-- coding: utf-8 --

The interpretation of the second line comment is the same as that used by Python - the engine searches for a match to the regular expression coding[=:]\s([-\w.]+) and then interprets the encoding part in the standard way (stripping '-' and normalizing to uppercase).

BidiDirection (7.0.0-rc-3)

The bidiDirection is a function that has been added to expose the engine's implementation of the Unicode Bidirectional Algorithm. It returns "ltr" or "rtl", depending on the computed base direction of the text it receives as a parameter.

Enable "umask" property on OS X (7.0.0-rc-3)

On POSIX systems, it is sometimes useful to set the umask when creating files or

directories. For example, this can be useful when creating temporary directories.

Previously, the "umask" property in LiveCode was only implemented on iOS, Linux

and Android platforms. It is now also available on Mac OS X.

File format change (7.0.0-rc-2)

In order to accommodate the saving and loading of unicode text throughout LiveCode, the file format of stacks has been changed. This means that stacks saved in 7.0 format cannot be opened in previous versions of LiveCode.

Legacy file formats are available to select when using the Save As... dialog. Saving in a legacy format will result in the loss of some information related to LiveCode 7.0, namely Unicode text in some areas (for example in object scripts), right-to-left formatting and tab alignment.

Array element pass by reference (7.0.0-rc-2)

It is now possible to pass parts of an array by reference. For example, the following

```
on mouseUp
local tArray
put "" into tArray[1][2]
passByRef tArray[1]
put tArray[1][2]
end mouseUp

on passByRef @rArray
put "changed" into rArray[2]
end passByRef
```

in the script of a button will result in "changed" appearing in the message box when the button is pressed.

This allows users to reduce the overhead associated with passing sub-arrays to handlers, as this would no longer require copying the sub-array internally.

Hebrew text is shown in reverse character order on Android (7.0.0-dp-7)

This bug fix involved incorporating the HarfBuzz library in Android builds. In addition to resolving bugs related to RTL text display, this has also enabled support for complex text shaping, so that combinations of characters in complex scripts such as Arabic are displayed correctly.

Unicode Support (7.0.0-dp-1)

Unicode and LiveCode

Traditionally, computer systems have stored text as 8-bit bytes, with each byte representing a single character (for example, the letter 'A' might be stored as 65). This has the advantage of being very simple and space efficient whilst providing enough (256) different values to represent all the symbols that might be provided on a typewriter.

The flaw in this scheme becomes obvious fairly quickly: there are far more than 256 different characters in use in all the writing systems of the world, especially when East Asian ideographic languages are considered. But, in the pre-internet days, this was not a big problem.

LiveCode, as a product first created before the rise of the internet, also adopted the 8-bit character sets of the platforms it ran on (which also meant that each platform used a different character set: MacRoman on

Apple devices, CP1252 on Windows and ISO-8859-1 on Linux and Solaris). LiveCode terms these character encodings "native" encodings.

In order to overcome the limitations of 8-bit character sets, the Unicode Consortium was formed. This group aims to assign a unique numerical value ("codepoint") to each symbol used in every written language in use (and in a number that are no longer used!). Unfortunately, this means that a single byte cannot represent any possible character.

The solution to this is to use multiple bytes to encode Unicode characters and there are a number of schemes for doing so. Some of these schemes can be quite complex, requiring a varying number of bytes for each character, depending on its codepoint.

LiveCode previously added support for the UTF-16 encoding for text stored in fields but this could be cumbersome to manipulate as the variable-length aspects of it were not handled transparently and it could only be used in limited contexts. Unicode could not be used in control names, directly in scripts or in many other places where it might be useful.

In LiveCode 7.0, the engine has been extensively re-written to be able to handle Unicode text transparently throughout. The standard text manipulation operations work on Unicode text without any additional effort on your part; Unicode text can now be used to name controls, stacks and other objects; menus containing Unicode selections no longer require tags to be usable - anywhere text is used, Unicode should work.

Adding this support has required some changes but these should be minor. Existing apps should continue to run with no changes but some tweaking may be required in order to adapt them for full Unicode support - this is described in the next section - Creating Unicode Apps.

Creating Unicode Apps

Creating stacks that support Unicode is no more difficult than creating any other stack but there are a few things that should be borne in mind when developing with Unicode. The most important of these is the difference between text and binary data - in previous versions of LiveCode, these could be used interchangeably; doing this with Unicode may not work as you expect (but it will continue to work for non-Unicode text).

When text is treated as binary data (i.e when it is written to a file, process, socket or other object outside of the LiveCode engine) it will lose its Unicode-ness: it will automatically be converted into the platform's 8-bit native character set and any Unicode characters that cannot be correctly represented will be converted into question mark '?' characters.

Similarly, treating binary data as text will interpret it as native text and won't support Unicode.

To avoid this loss of data, text should be explicitly encoded into binary data and decoded from binary data at these boundaries - this is done using the **textEncode** and **textDecode** functions (or its equivalents, such as opening a file using a specific encoding).

Unfortunately, the correct text encoding depends on the other programs that will be processing your data and cannot be automatically detected by the LiveCode engine. If in doubt, UTF-8 is often a good choice as it is widely supported by a number of text processing tools and is sometimes considered to be the "default" Unicode encoding.

New & Existing apps – things to look out for

- When dealing with binary data, you should use the **byte** chunk expression rather than **char** - **char** is intended for use with textual data and represents a single graphical character rather than an 8-bit unit.
- Try to avoid hard-coding assumptions based on your native language - the formatting of numbers or

the correct direction for text layout, for example. LiveCode provides utilities to assist you with this.
- Regardless of visual direction, text in LiveCode is always in logical order - word 1 is always the first word; it does not depend on whether it appears at the left or the right.
- Even English text can contain Unicode characters - curly quotation marks, long and short dashes, accents on loanwords, currency symbols...

New Commands, Functions & Syntax

Chunk expressions: byte, char, codepoint, codeunit

byte *x* **to** *y* **of** *text* -- Returns bytes from a binary string
char *x* **to** *y* **of** *text* -- As a series of graphical units
codepoint *x* **to** *y* **of** *text* -- As a series of Unicode codepoints
codeunit *x* **to** *y* **of** *text* -- As a series of encoded units

A variety of new chunk types have been added to the LiveCode syntax to support the various methods of referring to the components of text. This set is only important to those implementing low-level functions and can be safely ignored by the majority of users.

The key change is that **byte** and **char** are no longer synonyms - a byte is strictly an 8-bit unit and can only be reliably used with binary data. For backwards compatibility, it returns the corresponding native character from Unicode text (or a '?' if not representable) but this behaviour is deprecated and should not be used in new code.

The **char** chunk type no longer means an 8-bit unit but instead refers to what would naturally be thought of as a single graphical character (even if it is composed of multiple sub-units, as in some accented text or Korean ideographs). Because of this change, it is inappropriate to use this type of chunk expression on binary data.

The **codepoint** chunk type allows access to the sequence of Unicode codepoints which make up the string. This allows direct access to the components that make up a character. For example, á can be encoded as (a,combining-acute-accent) so it is one character, but two codepoints (the two codepoints being a and combining-acute-accent).

The **codeunit** chunk type allows direct access to the UTF-16 code-units which notionally make up the internal storage of strings. The codeunit and codepoint chunk are the same if a string only contains unicode codepoints from the Basic Multilingual Plane. If, however, the string contains unicode codepoints from the Supplementary Planes, then such codepoints are represented as two codeunits (via the surrogate pair mechanism). The most important feature of the 'codeunit' chunk is that it guarantees constant time indexed access into a string (just as char did in previous engines) however it is not of general utility and should be reserved for use in scripts which need greater speed but do not need to process Supplmentary Plane characters, or are able to do such processing themselves.

The hierarchy of these new and altered chunk types is as follows: **byte** *w* of **codeunit** *x* of **codepoint** *y* of **char** *z* of **word**...

Chunk expressions: paragraph, sentence and trueWord

The **sentence** and **trueWord** chunk expressions have been added to facilitate the processing of text, taking into account the different character sets and conventions used by various languages. They use the ICU library, which uses a large database of rules for its boundary analysis, to determine sentence and word breaks. ICU word breaks delimit not only whitespace but also individual punctuation characters; as a result the LiveCode **trueWord** chunk disregards any such substrings that contain no alphabetic or numeric characters.

The **paragraph** chunk is identical to the existing **line** chunk, except that it is also delimited by the Unicode paragraph separator (0x2029), which reflects paragraph breaking in LiveCode fields.

The hierarchy of these new chunk types is as follows: **trueword** *v* of **word** *w* of **item** *x* of **sentence** *y* of **paragraph** *z* of **line**...

Synonym: segment

The **segment** chunk type has been added as a synonym to the existing **word** chunk. This in order to allow you to update your scripts to use the newer syntax in anticipation of a future change to make the behaviour of the **word** chunk match the new **trueWord** behaviour.

We would anticipate changing the meaning of **word** with our 'Open Language' project. It requires us to create a highly accurate script translation system to allow old scripts to be rewritten in new revised and cleaner syntax. It is at this point we can seriously think about changing the meaning of existing tokens, including **word**. Existing scripts will continue to run using the existing parser, and they can be converted (by the user) over time to use the newer syntax.

Property: the formSensitive

set the **formSensitive** to false -- Default value

This property is similar to the **caseSensitive** property in its behaviour - it controls how text with minor differences is treated in comparison operations.

Normalization is a process defined by the Unicode standard for removing minor encoding differences for a small set of characters and is more fully described in the **normalizeText** function.

Command: open file/process/socket ... for <encoding> text

open file *"log.txt"* **for utf-8 text read** -- Opens a file as UTF-8

Opens a file, process or socket for text I/O using the specified encoding. The encodings supported by this command are the same as those for the **textEncode** / **textDecode** functions. All text written to or read from the object will undergo the appropriate encoding/decoding operation automatically.

Functions: textEncode, textDecode

textEncode(*string*, *encoding*) -- Converts from text to binary data
textDecode(*binary*, *encoding*) -- Converts from binary data to text

Supported encodings are (currently):

- "ASCII"
- "ISO-8859-1" (Linux only)
- "MacRoman" (OSX only)
- "Native" (ISO-8859-1 on Linux, MacRoman on OSX, CP1252 Windows)
- "UTF-16"
- "UTF-16BE"
- "UTF-16LE"
- "UTF-32"
- "UTF-32BE"
- "UTF-32LE"
- "UTF-8"

- "CP1252" (Windows only)

Spelling variations are ignored when matching encoding strings (i.e all characters other than [a-zA-z0-9] are ignored in matches as are case differences).

It is very highly recommended that any time you interface with things outside LiveCode (files, network sockets, processes, etc) that you explicitly **textEncode** any text you send outside LiveCode and **textDecode** all text received into LiveCode. If this doesn't happen, a platform-dependent encoding will be used (which normally does not support Unicode text).

It is not, in general, possible to reliably auto-detect text encodings so please check the documentation for the programme you are communicating with to find out what it expects. If in doubt, try "UTF-8".

Functions: numToCodepoint, codepointToNum

numToCodepoint(*number*) -- Converts a Unicode codepoint to text
codepointToNum(*codepoint*) -- Converts a codepoint to an integer

These functions convert between the textual form of a Unicode character and its numerical identifier ("codepoint"). Codepoints are integers in the range 0x000000 to 0x10FFFF that identify Unicode characters. For example, the space (" ") character is 0x20 and "A" is 0x41.

The codepointToNum function raises an exception if the argument contains multiple codepoints; it should generally be used in the form:

```
codepointToNum(codepoint x of string)
```

The numToCodepoint function raises an exception if the given integer is out of range for Unicode codepoints (i.e if it is negative or if it is greater than 0x10FFFF). Codepoints that are not currently assigned to characters by the latest Unicode standard are not considered to be invalid in order to ensure compatibility with future standards.

Functions: numToNativeChar, nativeCharToNum

numToNativeChar(*number*) -- Converts an 8-bit value to text
nativeCharToNum(*character*) -- Converts a character to an 8-bit value

These functions convert between text and native characters and are replacements for the deprecated **numToChar** and **charToNum** functions.

As the "native" character sets for each platform have a limited and different repertoire, these functions should not be used when preservation of Unicode text is desired. Any characters that cannot be mapped to the native character set are replaced with a question mark character ('?').

Unless needed for compatibility reasons, it is recommended that you use the **numToCodepoint** and **codepointToNum** functions instead.

Function: normalizeText

normalizeText(*text*, *normalForm*) -- Normalizes to the given form

The **normalizeText** function converts a text string into a specific 'normal form'.

Use the **normalizeText** function when you require a specific normal form of text.

In Unicode text, the same visual string can be represented by different character sequences. A prime example of this is precomposed characters and decomposed characters: an 'e' followed by a combining acute character is visually indistinguishable from a precombined 'é' character. Because of the confusion that can result, Unicode defined a number of "normal forms" that ensure that character representations are consistent.

The normal forms supported by this function are:

- "NFC" - precomposed
- "NFD" - decomposed
- "NFKC" - compatibility precomposed
- "NFKD" - compatibility decomposed

The "compatibility" normal forms are designed by the Unicode Consortium for dealing with certain legacy encodings and are not generally useful otherwise.

It should be noted that normalization does not avoid all problems with visually-identical characters; Unicode contains a number of characters that will (in the majority of fonts) be indistinguishable but are nonetheless completely different characters (a prime example of this is "M" and U+2164 "M" ROMAN NUMERAL ONE THOUSAND).

Unless the **formSensitive** handler property is set to true, LiveCode ignores text normalization when performing comparisons (is, <>, etc).

Returns: the text normalized into the given form.

```
set the formSensitive to true
put "e" & numToCodepoint("0x301") into tExample -- Acute accent
put tExample is "é" -- Returns false
put normalizeText(tExample, "NFC") is "é" -- Returns true
```

Function: codepointProperty

codepointProperty("A", "Script") -- "Latin"
codepointProperty("β", "Uppercase") -- false
codepointProperty("σ", "Name") -- GREEK SMALL LETTER SIGMA

Retrieves a UCD character property of a Unicode codepoint.

The Unicode standard and the associated Unicode Character Database (UCD) define a series of properties for each codepoint in the Unicode standard. A number of these properties are used internally by the engine during text processing but it is also possible to query these properties directly using this function.

This function is not intended for general-purpose use; please use functions such as toUpper or the "is" operators instead.

There are many properties available; please see the version 6.3.0 of the Unicode standard, Chapter 4 and Section 5 of Unicode Technical Report (TR)#44 for details on the names and values of properties. Property names may be specified with either spaces or underscores and are not case-sensitive.

Examples of supported properties are:

- "Name" - Unique name for this codepoint
- "Numeric_Value" - Numerical value, e.g. 4 for "4"

- "Quotation_Mark" - True if the codepoint is a quotation mark
- "Uppercase_Mapping" - Uppercase equivalent of the character
- "Lowercase" - True if the codepoint is lower-case

Updated Functions

Function: binaryEncode

A new letter has been introduced to allow one to binary encode unicode strings.
Following the dictionary definitions, it consists of:

u{<encoding>}: convert the input string to the encoding specified in the curly braces, and output up to amount bytes of the string created - stopping at the last encoded character fitting in the amount - padding with '\0'.

U{<encoding>}: convert the input string to the encoding specified in the curly braces, and output up to amount bytes of the string created - stopping at the last encoded character fitting in the amount - padding with encoded spaces, and then '\0' if the last encoded space cannot fit within the amount specified.

The encoding, surrounded by curly braces, is optional - no one specified would default to the behaviour of 'a' - and must match one of those applicable to textEncode

Function: binaryDecode

A new letter has been introduced to allow one to binary decode unicode strings.
Following the dictionary definitions, it consists of:

u{<encoding>}: convert amount bytes of the input string to the specified encoding, padding with '\0'.

U{<encoding>}: converts amount bytes of the input to the specified encoding, skipping trailing spaces.

The encoding, surrounded by curly braces, is optional - no one specified would default to the behaviour of 'a' - and must match one of those applicable to textEncode

Deprecated Features

Functions: numToChar, charToNum

These functions should not be used in new code as they cannot correctly handle Unicode text.

Property: useUnicode

This property should not be used in new code, as it only affects the behaviour of **numToChar** and **charToNum**, which are themselves deprecated.

Functions: uniEncode, uniDecode

These functions should not be used in new code as their existing behaviour is incompatible with the new, transparent Unicode handling (the resulting value will be treated as binary data rather than text). These functions are only useful in combination with the also-deprecated unicode properties described below.

Function: measureUnicodeText

This function should not be used in new code. **measureUnicodeText**(*tText*) is equivalent to

measureText(**textDecode**(*tText*, "UTF16")).

Properties: unicodeText, unicodeLabel, unicodeTitle, unicodeTooltip, unicodePlainText, unicodeFormattedText

These properties should not be used in new code; simply set the text, label, title etc. as normal. Assigning values other than those returned from **uniEncode** to these properties will not produce the desired results.

The following are now equivalent:

```
set the unicodeText of field 1 to tText
set the text of field 1 to textDecode(tText, "UTF16")
```

and similarly for the other unicode-prefixed properties.

Specific bug fixes (7.0.6-rc-3)

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

	Make rawKeyUp and rawKeyDown consistent on non-Latin chars
15516	**Don't crash when attempting to get a child of a non-group control**

Specific bug fixes (7.0.6-rc-2)

15526	Trailing 'break point' in the points of a graphic not preserved.
15482	Fix Linux drag-and-drop issues between LiveCode controls

Specific bug fixes (7.0.6-rc-1)

15474	Max interger literal number limited to 2^32 on 64-bit Linux
15467	Memory leak in MCPlayer::resolveplayerfilename
15455	custom property gets truncated if it contains a NULL byte
15424	LC 7 not playing file from server in Windows
15422	The number of trueWords returns the number of words as per old definition
15403	mobileAddContact doesn't work
15386	Failure to incorporate code with 'include' on 7.0 servers
15383	global allowabledragactions property broken
15371	mobileGetContactData fails when there are multiple labels of the same property
15370	mobileFindContact crashes LiveCode when contact not found
15368	put before line x of URL broken on LC 7 & 8
15359	Export image with list broken in LC 7.0
15352	Set label field contents to empty in Property Inspector causes crash
15327	offset(tosearch
15315	All text properties of a field's text are deleted when putting text after the last line
15314	"secure socket" crashes livecode
15305	Entering keystroke option-e twice in a field crashes LiveCode
15175	tabAlign field property is not saved in the stack file
15040	alt+0244 text entry field does not work on Windows
14792	MobileUpdateContact fails on iOS/Android in LC 7.0.x
14766	iOs standalone encryption not working in LiveCode 7.0

3537 When inputting non-latin data into a field

Specific bug fixes (7.0.5-rc-3)

15309 token chunk broken in complex chunk expressions
15264 Crash in Stack with Player Object when selecting "Tracks" in the player property inspector
15258 Custom properties lose unicode text when copied
15243 Checking tabStops property over line range with differing tabStops set causes crash
15241 Crash when setting clipboardData to wrong type of value
15151 libJson not working on Android sim or Device
15124 mobilegetcontactdata crashes app on ios 8.2
14994 Clicking in field with listBehavior true and lockText false crashes
14438 mobileGetContactData does not work on Android in LC 7.0

Specific bug fixes (7.0.5-rc-1)

15296 mobileExportImageToAlbum fails on Android
15267 byte x to - 1 of string fails in 7.0.4
15196 Setting Filename of Player in Substack to Empty Causes Crash
15193 export snapshot leaks memory and eventually crashes LC 7.0.4 rc3
15168 Setting the beep sound crashes iOS simulator in LC 7.0.x
15161 WeekDayNames on Linux Ubuntu miss accented characters
15139 Putting string data into complex byte chunk crashes LiveCode
15123 Crash when deleting a control in edit group mode
15105 Appleevent handler crashes 7.0 standalone
15090 App quits with setting accelerated rendering on opening in iphone 5S and 6
15030 the effective textstyle crashes LiveCode
14835 Fix word wrapping in fields containing tabs
14717 Points properties of graphic are too strict - they should except either '
14700 revZipOpenArchive can fail on 64-bit Linux
14696 Open File Refuses To Work With 1GB+ Files on Windows
14413 revZipOpenArchive fails in 7.0.1 with Unicode characters
14287 openSockets returns an extra char to an open port

Specific bug fixes (7.0.4)

15164 Externals do not load if their path has a Unicode character

Specific bug fixes (7.0.4-rc-3)

15140 String list parsing for properties incorrect
15062 Standalones saved in a folder with Unicode char will not launch on Mac
15045 Parse issue with nested char chunk expression
15034 import snapshot does not work correctly on 64-bit Linux.
14965 setting cursor to id of non-existing image crashes LC 7.0.4 rc1
14960 LC8 DP1 crash while importing JPG image
14885 'sort cards of background ...' crashes
14855 mobileSensorReading(sensor_name
14833 Format strings of the form "%0s" do not work correctly.

14814 print cardâ€¦ into pageRect fails in LC7.0.3
14712 Save As converts names to lower-case on Linux
14611 answer file/folder .. with .. does not work anymore with backslash delimiter
14326 revZip externals does not work on iOS

Specific bug fixes (7.0.4-rc-2)

14139 ChartsEngine 1.2.1 can't be registered with LiveCode 7

Specific bug fixes (7.0.4-rc-1)

14661 LiveCode 7 IDE Doesn't Use Standard Windows Font
14658 Recursive array ops fail debug mode mutability assertion
14612 read from file COM1: --- port doesn't work ---
14604 [LC 7] iOS mobile player 'loadstate' property never contains 'playthrough'
14587 LC 7 Server buffers output
14536 Return parameters require a local variable to be defined first
14531 EOF is returned when reading exactly the number of chars in a file
14515 put URL into URL"binFile:eXample.mp3" crash on Android
14504 Strings which have been 'unnativized' by the engine lose the CanBeNative flag when copied
14480 Typo in textEncode dictionary entry
14477 Trying to get $_FILES array fails since server version 7.
14467 Unable to get line Attributes of Char Chunks in LC 7
14463 Second invocation of play command causes crash on iOS
14462 Crash related to shell command on Linux
14457 Segment-based URIs yield an error since server version 7
14441 Shell command takes much longer to run in LiveCode Server 7 than in 6
14439 Minus sign ignored when assigning value to variable at declaration
14437 Clearing a Player filename causes LC to crash.
14383 Android standalones can not open a new file for update on the device
14160 LiveCode 7.0 silently crashes if the Print Spooler is turned off

Specific bug fixes (7.0.2-rc-2)

14449 Player Objects in 7.0.2 (rc1) don't work
14428 set the background color of a field from a different card crashes live code
14424 Getting 'the fontfilesinuse' can cause a crash.
14412 The detailed files does not return urlencoded filenames
14409 Reading from socket crashes LiveCode
14401 [[In-App Purchase]] mobileStorePurchaseError returns empty in LC 7.0
14399 setting the bg color of a grc that has a fillgradient in 7 does not wipe gradient
14386 Evaluating 1.884956+0 and then using in string context can make it not convert back to a number.
14384 put +1 into variable causes LC to crash
14378 Crash when pasting text
14377 "repeat for i = 1 to 5" is not a syntax error
14376 hang/crash on quit LC 7.0.1 when referencing owner of me in script
14354 Building strings bigger than memory by concatenation crashes LiveCode

14305 Print corruption in LiveCode 7
14229 Fields without fixedHeightLine do not refresh when the last char of a line is deleted
14057 internal API error (2) when using external

Specific bug fixes (7.0.2-rc-1)

14343 mobileStoreMakePurchase misbehaving in LC7.0.x
14330 openstacks() crashes when called in the shutdown handler
14304 launch URL is broken for (space)
14303 Problem saving stack with custom properties
14302 Recording does not produce file in correct location
14278 Unicode PDF file name changes to ??????.pdf
14277 Crash when any of the 'repeat with' expressions triggers an error
14259 Externals fail to load when livecode server script is started via "#!/usr/bin/env livecode-server"

Specific bug fixes (7.0.1)

14227 Saving stack in legacy format causes crash
14220 shell() function has a minimum 0.25 second overhead
14219 HKEY_CURRENT_USER returns emtpy
14203 uniencode / decode broken
14181 Server does not return from hostNameToAddress
14161 V7.0.1-rc3 IDE is slow

Specific bug fixes (7.0.1-rc-4)

14208 Number of bytes != number of chars when converting non-native string to data
14170 hilitedbutton fn changed
14162 rawkey messages not sent hwne using russian input
14154 LC7 effective colors are reported as empty
14149 tabbed data not correctly showing when stack was save in LC 7.0.1 RC2
14147 String are sometimes wrongly considered as empty arrays
14120 Date/Time Pickers on Android Crashes when Cancelled
14001 LiveCode Server on Linux not recognizing revOpenDatabase
13591 Set default value for the "shellCommand" on LiveCode Mac OS X server.

Specific bug fixes (7.0.1-rc-3)

14135 read from process X until empty slows down the IDE
14123 convert to seconds fills "the result" with the seconds value
14110 Cannot open file on a file of size zero
14092 Can't set dropShadow knockout or glow range
14088 put the templateXXXXX crashes LiveCode
14076 Sheet stack .... in stack broken
14068 MobileBusyIndicatorStart crashes iOS app
14064 Errant ` character in htmlText
14062 mobileControlDo execute does not function in LiveCode 7 and can result in hard crash
14060 Avoid duplicate '..' entries in 'the folders' when not on OS X.

14059 The 'format' function does not produce a '0' padded result
14058 incorrect parameters sent to the relaunch Handler on Windows
14055 Effective patterns return random numbers
14053 Error align text from Text menu
14052 LiveCode 7 crashes without error in many commands
14049 setting the backgroundcolor of a line in a field no visible reaction
14046 Native Android scroller fails in 7.0.1rc1 but works in 6.6.2
14044 Do ... in caller doesn't work in LC 7
14043 Stacks copied in a standalone are not accessible on Mac
14032 revZip library doesn't support Unicode paths or Unicode callbacks
14031 mobilePixelDensity returns empty on iOS
14029 Read server scripts correctly when they cannot be memory-mapped.
14026 LCObjectGet is crashing
14016 mobileGetContactData Unreliable
14015 urldecode and urlencode broken with Unicode characters
14006 Saving a print job as pdf is not treated the same under LC6 and LC7
13993 Wrong name return by a dragdrop for a path with accented letters
13988 put data after or before variable should result in data if variable is empty
13954 Can't set the currentTime of player
13948 Cursor splits in 2 in text field
13865 Automatic Licensing of marketplace products does not work in 7.0
13695 Correctly handle '..' as 2nd path element when simplifying relative paths.

Specific bug fixes (7.0.1-rc-2)

13987 Putting into bytes takes a long time
13972 open card treats string as variable breaking old code
13943 Getting centerRect returns incorrect rectangle
13934 Setting the colour of a line is not displayed instantly
13930 Execution Error in tmControl frontscript
13925 getting a color property from a chunk always returns 'mixed'
13920 [[Player]] loadedTime property not supported in LC 7.0
13915 Canceling an open printing does not work
13913 Print this cd from... to... doesn't work
13907 Small script using 'type' crashes LC 7.0.1
13902 It can't be used as a parameter name
13897 the formattedleft of the clickChunk not working properly in LiveCode 7.0 any more
13893 ask file with type causes a crash
13890 Empty array is not empty
13887 Fix some internal oddities in the field control
13879 Saving a field with metadata corrupts the stack file
13877 Download of mp3 file creates unplayable file
13876 The format() can not format the Unicode string correctly.
13866 revPrintText prints half of the text on Windows
13853 Update dragData docs
13851 Error when saving standalone

3680 Initializers for constants and locals too strict

Specific bug fixes (7.0.1-rc-1)

start using throws inappropriate error if stack script won't compile
13850 ask file and ask folder default to wrong folder
13840 Stopped MCS_savebinaryfile() from failing if MCresult is non-empty.
13833 Rendering many grcs when acceleratedRendering is true crashes hard
13827 Can't use double-byte characters for a standalone name
13824 Hang in MCExecContext::TryToEvaluateExpressionAsNonStrictBool()
13821 Unable to set cursor hotspot
13809 Specifying "width" in measureText throws error
13804 export snapshot crashes LC7GM 1
13803 Potential instability after setting defaultNetworkInterface.
13802 Android standalone big text collapse
13797 Using put into url causes executable bit to be set on the target file if it is created on Linux.
13796 Creating a script only stack causes unwanted messages to stderr.
13791 Crash when saving a script-only stack.
13790 iPhone player object doesn't respect the rect property on creation
13787 'ends with' can return the wrong result if a string contains surrogates or combining sequences.
13785 The 'properties' property returns different keys from the 6.7 version.
13784 Crash exporting image as png
13783 Empty is among the items of empty
13780 mobileSetAllowedOrientations broken on android
13778 Setting the httpHeaders causes crash on "put url"
13776 [[LC 7.0]] Player plays audio but not video
13770 Every string contains empty
13751 Player Volume Control Erratic
13730 Cannot specify text encoding of server script file.
13704 securityPermissions not working for disk and network access
13554 Corruption of stacks

Specific bug fixes (7.0.0)

13763 Native chars don't hash to the same value as equivalent unicode chars
13761 Fix UTF-8 output from server scripts
13757 The detailed files is wrong on Windows.
13745 answer file with type treats empty filter as wild
13742 Export "the styledText" runs as "text" rather than "unicodeText"
13741 Fix multiple middle-click pasting issues
13740 numToByte outputs text rather than data
13738 audioClip references not being resolved correctly
13737 "is an array" is only true if there is at least one key
13736 charToNum(empty) should return empty
13732 Saving a stack with a binary string in a custom property in 5.5 format causes truncation at 65535 bytes.
13728 Issue with externals and reading values from LiveCode variables

13727 Always insert a linebreak after vtab
13725 Various fixes to binaryDecode
13724 Strings should convert to empty arrays
13717 Link Colors Inconsistent
13684 hidePalettes property defaults to false
13658 Data corrupted by the shell() function on server
13450 Independence resolution does not work well with a Browser Object

Specific bug fixes (7.0.0-rc-3)

13706 Fix a bug in the image saving code causing stackfile corruption
13680 item delimiter not deleted in target string if longer than 1 character
13674 Implement diskSpace function on Linux.
13671 repeat for each item or line does not use multichar delimiter
13664 Livecode 7.0 rc 2 does not sort certain strings correctly when the numeric form is used
13660 Crash (SIGSEGV) during drag & drop operation
13642 Improved documentation for the umask property.
13626 Android app crashes when back button pressed for a second time
13619 Setting a non-readable default folder makes 'the folders' fail
13610 dragImage with id upper than 65535
13605 'set the clipboardData' can cause crashing on Windows
13594 Evaluation faulty
13587 Stacks with Unicode filename won't open from the Dock
13579 Behaviour for 'there is a url...' in LiveCode 7.0 inconsistent with previous versions
13559 firstindent can not be set via styledText array in 7.0 RC2
13553 Crash when putting one image into another
13552 Crash navigating to a card
13550 Deleting word chunk erroneously removes preceding whitespace
13548 Fix calculation for tab-on-return
13542 A card with many (>80) text fields causes a crash on android device when made visible
13534 Fix string -> bool conversion in the v1 externals interface
13530 revXMLCreateTreeFromFile does not work with decomposed accented characters in filename
13529 Setting TabWidth on a lines of a field crashes with runtime error
13528 Parentheses appear in disabled tab menu items
13511 Fix a typo in the Win32 time formatting code
13509 Livecode 7 remembers cleared block attributes
13499 Fix "answer file" opening in wrong folder
13496 Fix a crash due to uninitialised locale on server
13480 Fix a potential nil pointer crash
13426 mobileDeviceOrientation() not working in LC 7.0.0(rc1)
13225 Linux: update engine mouse coords on click events
12786 \+\ key combination now works on Linux desktop.
12545 Fix I/O for serial devices
12444 Fix Windows command line parsing

Specific bug fixes (7.0.0-rc-2)

13473	Add image area to card and in the property inspector put a URL as the source
13467	PrintPageNumber returns -1 by default
13465	More memory leaks in handler parameter creation
13463	Fix deployment to Windows from 64-bit Linux
13461	sort removes textColor in 7.0 RC1
13460	'convert' output is incorrectly formatted
13454	Memory leaks in handler parameter creation
13453	variable watch not working
13444	Make sure data is sent when doing POST or PUT from LiveCode Server.
13437	Setting the currentTime of player result in error statement "Not a number" in LC 7.0
13433	No mention of file format change in v7 Release Notes
13430	Nudging an object with arrow keys is broken
13428	Play stop not working
13422	setting iconGravity needs redraw
13407	Hilite artifact on the last column in VGrid mode
13405	IDE Menu Shortcuts only work when menu is open
13401	Setting a button label to empty when it has one already does not reinstate button name as label
13400	Severe slowdown in copying blocks of bytes / chars
13394	go next marked cd does not work anymore
13388	Put text after a buttons text causes LC7 RC1 to crash
13378	'get' causes a crash on server
13375	setting tabWidths results in erroneous tabstops in LC7RC1
13361	Script editor replaces *more* than the selection with new text
13359	matchText can't assign values to variables that are parameters
13356	empty not among the items of a list with a trailing comma
13353	Getting htmltext when there is firstindent results in LiveCode crashing
13352	Crash when sorting lines of a field
13348	RawKeyUp and KeyUp are sent twice in LC 7
13346	LineOffset should return 0 in LC7 RC1
13340	Keys and values of are corrupted in LiveCode 7
13336	mobilePixelDensity returns 100+ digit number
13335	set the textFont crashes Android
13332	Field allows line break on non-breaking space
13329	Cannot import photo into stack on Android device.
13323	'there is a file ' is not always right
13316	Setting line chunk properties on multiple lines doesn't work
13315	textDirection does not survive save & load
13314	Inconsistent Line breaks using html text
13312	Setting read-only global properties crashes Livecode
13311	Flagged block index incorrect
13300	'Set the menubar to ' causes crash
13297	Combine by Column broken in LC7DP10
13296	"the processor" returns "arm" on RaspberryPi
13294	Fix a Linux MPlayer crash
13289	Set the statusiconmenu - when used

13288 controlAtScreenLoc always returns the card
13276 abbreviated name isn't understood anymore
13263 deleted field text visible in lc7
13259 Fix system time formatting on Windows
13258 null after file name in lc7 drag drop
13255 Script debugger points to empty script when unknown XML parse error occurs
13249 tabbed data in list mode does not hilite hilitedLine correctly LCDP10
13247 Setting large htmltext is very slow
13239 iOS hard crash when using encyption
13219 Crash in OSX locale caching
13214 Hang when creating a player
13204 effective hiliteColor has changed behaviour in LC7DP9
13200 LC7 cannot save the title of stack
13186 Name comparison failure when using menuPick from tab panel
13179 Crash when getting mac resources
13042 Alt- combinations don't generate the correct character.
12903
12776 'The number of elements of tVar' for non-array tVar hangs LC7
12547 Make arrayEncode encode in 7.0 format by default
12539 Don't draw tab characters
12502 Fix a null-pointer deref in PDF printing
11971 Password protected stacks are corrupted by LiveCode 7

Specific bug fixes (7.0.0-dp-10)

13178 Player won't play from server
13177 start using fails in livecode 7 server
13176 core image visual effects broken in LC7DP9
13146 Print to PDF fails in 7DP9
13145 ImageData display by reference hangs 7DP9
13144 answer files behaviour is broken in 7DP9
13143 LC7dp9 replaces mainStack name with /Applications in Save As dialog
13139 Incorrect parsing of
13135 Ensure that setting or getting custom properties with an index triggers the appropriate SetProp/GetProp
13124 cursor split in certain conditions in tabbed data field
13108 text selection in columnar data incorrect
13106 tabbed text with vGrid on in right align or centered mode flows over to the left
13077 Setting htmltext of field chunks can cause unexpected block order switching

Specific bug fixes (7.0.0-dp-9)

13122 Break stopped working in if statements within switch
13115 [[player]] player missing formattedwidth and formattedheight properties
13103 option
13100 LC7 DP8 Combo box label anomoly
13097 Image with no filename is not blank

13090 LC7 DP8 Split by column fails to honour blank lines
13089 Setting text of a combobox does not set the label
13084 LiveCode crashes when selecting PDF printer in printer dailog Windows desktop
13082 imageSource sometimes can't be deleted
13081 Prevent crash when evaluating non-container chunk
13079 select before | after text selects all text of field
13076 text in field does not change color when textColor property is set
13070 Fix a pointer cast that broke copy-and-paste in 64-bit builds
13057 Unable to change to initial orientation after changing orientation of device
13056 arrayDecode no longer throws an error on invalid input
13050 arrayDecode causes error when encoded array contains binary elements
13043 Stack gets corrupted after removing it from memory
13027 System icon shows rather than LiveCode icon when changing application
9058 Unmaximise windows on Linux if the max width/height is exceeded
8637 Make the "hidepalettes" property work on Linux

Specific bug fixes (7.0.0-dp-8)

13029 Windows statusiconmenu not parsed correctly
13024 Launch URL fails to launch text documents
13022 Clear Linux backdrop window after changing background colour
13018 Split by ~~and is broken with Unicode~~
12998 "Exit" is too in menu "File" on Mac
12984 setting the callback of a player crashes LiveCode
12983 Crash when looking for qteffects
12981 Clear "transient for" hint when clearing Linux backdrop
12972 Player filename dialog does not allow audio files to be selected
12952 tabbed date incorrectly displayed when vertical lines on
12951 text selection in tabbed text inconsistent
12948 Crash when opening custom property inspector having a property with more than 65535 bytes
12945 Problems with tabStops property
12937 param() is not parsed
12936 Video player crash when setting callbacks
12931 Prevent Linux backdrop from gaining focus
12925 Text - > Align does nothing
12924 Setting the style
12921 Install 32-bit and 64-bit Linux engines to different paths
12918 Object -> Flip Image on an image with a filename crashes
12916 Closing the Page Setup dialog causes a crash
12910 Script editor crashes
12909 Fix a crash on Linux when taking a snapshot of the screen
12907 File > Import as control > Snapshot from screen
12905 Set Linux geometry hints on window creation
12901 Object colors not selectable in inspector
12896 Cursor navigation broken in tabbed fields
12893 Crash when dragging away from player icon in Tools palette

12874 revBrowser (both original and CEF) crashes LiveCode 7.0 DP7
12847 Property inspector's selection menu is broken
12846 Property Inspector updates too often when moving a control
12843 thumposition returns decimal value in LC7 dp6
12729 Token chunk expression is not allowing for quotes correctly
12162 Inconsistent handling of PS in 'put into' and 'put after'

Specific bug fixes (7.0.0-dp-7)

12823 Selecting subsequent cells in a tabbed field results in incorrect highlighting
12814 Setting textDirection should force field recalculation
12797 filter with regex not working
12795 'The number of elements of tVar' for non-array tVar hangs LC7
12792 Pasting text from Text Edit into field creates gibberish
12790 Ctrl-m does not close the message box
12789 Clicking on stack listed in Application Browser causes crash
12778 Double clicking in the script editor doesn't highlight words
12777 Copy command crashes in release mode
12733 Error when getting or setting char chunk properties of buttons
12721 keyUp keyname returns gibberish
12700 Launch URL not working on LC7 in Android and iOS emulators
12697 Setting tabStop less than the preceding one on a field causes text to overlap
12695 Android video does not display
12676 Adding number to numeric value in variable gives incorrect result on LC7
12672 LC 7.0DP6 Crash on Save After Editing Large Script
12659 Error on Android when reading files list from the stack folder path
12656 Decomposing native strings doesn't work
12651 back key can not work
12650 Copying externals files to android app fails
12644 Filtering unicode text with wildcard can result in false positives
12610 Split by column causes crash
12596 Number of controls of card returns wrong value if given a card id
12595 Printing to PDF does not yield all information
12576 drawing_bug_when_rotating_graphic
12574 REGEX : matchText result not as expected
12562 Changing the back color of a line which contains a tab makes LC crash
12552 go to url internet stack path does not work
12540 Clipboarddata should return utf16 data for 'unicode' mode
12538 Read from process until empty
12532 Adding a new element to an array can be very slow
12515 crash on clicking linktext (on second click)
12488 Tabbed characters are cut off on the left
12478 Retrieving data from url results in garbled data on iOS from LiveCode 7
12477 Native mobile controls created with mobGui do not seem to function under LiveCode 7.0
12343 Hebrew text is shown in reverse character order on Android
12166 Fix cursor movement over zero-width characters

Specific bug fixes (7.0.0-dp-6)

12544 send command with a parameter which contains a quote breaks param parsing
12530 embedded wav sound crashes Project Browser and Properties inspector in LC 7 dp5
12527 paragraph chunk returns empty when string does not include end of paragraph mark
12521 Fix highlights for non-left-aligned lines in fields
12517 Quicktime using stacks crash on open
12514 dragData with a private content extracted from a string by using a chunk keyword (word
12511 charIndex property missing
12510 setting stack decoration errors
12509 fullscreenMode "showAll" breaks IDE
12493 open file for binary read/write erroneously converting line endings

Specific bug fixes (7.0.0-dp-5)

12499 trueWord n + m of tText for n the number of trueWords of tText always returns trueWord n
12497 pageRanges property missing from LiveCode 7.0
12496 [[Bugfix 12496]] Set the clipping rectangle for text blocks correctly
12494 Setting the randomSeed to large number fails in 7.0
12491 "Go to Definition" doesn't work in script editor
12489 filter/replace difference in 7.0
12486 [[Bugfix 12486]] Add missing MovieControllerID property to the Player property table
12483 Graphic effects not working in 7.0 DP4
12482 replace does not work
12074 Answer dialog messages should be aligned to the right

Specific bug fixes (7.0.0-dp-4)

12459 Setting any graphic effects to "none" crashes LC 7 dp3
12457 sorting marked cards with single unmarked card crashes LiveCode
12432 clickchunk and click text are not identical
12428 Lc 7.0 DP3 does not sanitize data when setting points of polygon
12423 If you choose the browse tool (run) after Editing a group - Livecode crashes.
12422 Sort puts a "p" after the last character and foreign letters is not sorted correct
12409 Fields in LC 7 fail to display binfile url imagesource
12407 'Garbage' with read from socket
12360 open file as utf-8 mode doesn't work exactly as documented
12345 AVD's appear in the list but can't be selected for testing.
12344 Can't open recent file
12309 Build for Windows fails with i/o error
12288 Prevent User Samples stack hanging due to resize error
12246 Serial I/O fails on write
12192 linux uninstaller needs execute permission
12061 Can't test an app on Android
11989 arrayDecode on a file containing the result of arrayEncode on an empty array causes execution error

Specific bug fixes (7.0.0-dp-3)

12290 saving 2.7 file format stack causes crash
12244 case sensitive does not work
12204 textEncode ASCII support is actually native
12195 equality testing is slow
12194 'char/byte/codepoint 1 of s' is slow
12184 'repeat for each byte b in empty' crashes
12180 'the number of bytes of ...' is slow
12179 Fetching byte chunks does not clamp the range to the bounds of the input data.
12168 Sometimes length() and number or chars are wrong
12160 Put after/before on an uninitialised
12150 LiveCode crashes when changing the window kind
12147 create button in group command fails
12143 The mousechunk end index is one larger than it ought to be
12140 Erroneous Socket Timeout Error
12138 the drawer command crashes Livecode 7.0 when using '...at position' variant.
12123 Fix wrong application title displaying on Linux
12122 Update GTK icon cache post-install
12118 revExecuteSQL writes incomplete data into SQLite BLOB columns
12078 Scrambled word order for label field with Hebrew and English Text
12075 Buttons that contain Hebrew Text is in wrong order
12007 Linux Standalone does not run. Segmentation fault.
11993 "save stack" corrupt password protected stacks
11979 IDE fails to launch when installed to a Unicode path
11973 char 1 of (e + combining acute accent) returns e
11962 Split command causes IDE to stop responding
11961 IDE takes 8 seconds when adding a new line in Script Editor
11941 repeat loop is very slow in 7.0 DP1
11939 Opening the TestFramework stack crashes LiveCode

Specific bug fixes (7.0.0-dp-2)

12104 Convert command fails with invalid date since 7.0
12097 setting acceleratorModifiers of button causes crash
12081 OSX picking wrong file extension for filenames with two '.' characters
12071 hiliteColor and borderColor is not working in 7.0DP1
12070 hGrid
12067 Group with label can't be saved in 5.5 file format
12065 formatting hex string crashes LiveCode 7.0
12042 New chunk types (paragraph
12038 'lock screen for visual effect in rect...' not working
11996 numToByte works differently form numToChar in 6.6
11985 put does not populate the result on iOS
11981 calling mobileControlTarget () crashes the application
11963 Dotted border of selection in List control is incorrectly aligned
11960 LC crashes when selecting wrapped text in Contents pane

11958 Text wrapping improperly breaks text mid-word
11954 sort field does not work
11953 sort card of stack crashes
11950 mark card does not work
11949 find string in field does not work
11948 Export snaphot crashes LiveCode when it should return empty rect error
11947 Vertical tabulation in a field causes the engine to hang
11945 The number of paragraphs reported value is incorrect
11943 Script Editor does not resize correctly with the resize handle
11940 Variables not being resolved in the script debugger.

IDE changes

Specific bug fixes (7.0.4-rc-1)

14373 Linux icon not checked when building for Linux64 or Linux ARM only
14348 Database drivers missing when building Linux 64-bit standalones

LiveCode Builder changes

Dictionary changes

- The entry for **export with palette** (*command*) has been updated.
- The entry for **specialFolderPath** (*function*) has been updated.
- The entry for **dateItems** (*keyword*) has been updated.
- The entry for **dontUseQT** (*property*) has been updated.

Previous Release Notes

7.0.4 Release Notes	http://downloads.livecode.com/livecode/7_0_4/LiveCodeNotes-7_0_4.pdf
7.0.3 Release Notes	http://downloads.livecode.com/livecode/7_0_3/LiveCodeNotes-7_0_3.pdf
7.0.1 Release Notes	http://downloads.livecode.com/livecode/7_0_1/LiveCodeNotes-7_0_1.pdf
7.0.0 Release Notes	http://downloads.livecode.com/livecode/7_0_0/LiveCodeNotes-7_0_0.pdf
6.7.4 Release Notes	http://downloads.livecode.com/livecode/6_7_4/LiveCodeNotes-6_7_4.pdf
6.7.2 Release Notes	http://downloads.livecode.com/livecode/6_7_2/LiveCodeNotes-6_7_2.pdf
6.7.1 Release Notes	http://downloads.livecode.com/livecode/6_7_1/LiveCodeNotes-6_7_1.pdf
6.7.0 Release Notes	http://downloads.livecode.com/livecode/6_7_0/LiveCodeNotes-6_7_0.pdf
6.6.2 Release Notes	http://downloads.livecode.com/livecode/6_6_2/LiveCodeNotes-6_6_2.pdf
6.6.1 Release Notes	http://downloads.livecode.com/livecode/6_6_1/LiveCodeNotes-6_6_1.pdf
6.6.0 Release Notes	http://downloads.livecode.com/livecode/6_6_0/LiveCodeNotes-6_6_0.pdf
6.5.2 Release Notes	http://downloads.livecode.com/livecode/6_5_2/LiveCodeNotes-6_5_2.pdf
6.5.1 Release Notes	http://downloads.livecode.com/livecode/6_5_1/LiveCodeNotes-6_5_1.pdf
6.5.0 Release Notes	http://downloads.livecode.com/livecode/6_5_0/LiveCodeNotes-6_5_0.pdf
6.1.3 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