**LiveCode**

The LiveCode programming language (formerly the "Revolution" programming language) is a commercial [cross-platform](http://en.wikipedia.org/wiki/Cross-platform) [rapid application development](http://en.wikipedia.org/wiki/Rapid_application_development) language inspired by [Hypercard's](http://en.wikipedia.org/wiki/Hypercard) programming language [HyperTalk](http://en.wikipedia.org/wiki/HyperTalk). The "Revolution" development system was based on the [MetaCard](http://en.wikipedia.org/wiki/MetaCard) engine technology which [Runtime Revolution](http://en.wikipedia.org/wiki/Runtime_Revolution) later acquired from MetaCard Corporation in 2003. The platform won the Macworld Annual Editor's Choice Award for "Best Development Software" in 2004. "Revolution" was renamed "LiveCode" in the fall of 2010. "LiveCode" is developed and sold by [Runtime Revolution](http://en.wikipedia.org/wiki/Runtime_Revolution) Ltd., based in Edinburgh, Scotland.

LiveCode runs on [iOS](http://en.wikipedia.org/wiki/IOS_%28Apple%29), [Android](http://en.wikipedia.org/wiki/Android_%28operating_system%29), [Mac OS X](http://en.wikipedia.org/wiki/Mac_OS_X), [Windows 95](http://en.wikipedia.org/wiki/Windows_95) through [Windows 7](http://en.wikipedia.org/wiki/Windows_7), and several variations of Unix, including Linux, Solaris, and BSD. It can be used for both desktop and server/CGI applications. The first version for iOS (iPhone and iPad) was released in December 2010. The first version to deploy to the Web was released in 2009.

LiveCode uses a high level, English-like programming language that is dynamically typed. The high-level programming language and compile-free workflow allow for code that is self-documenting and easy for casual programmers to comprehend.

-from Wiki

**General Terms:**

**Local Notification –** Local notifications allow applications to communicate with the user, whether they are running in the foreground, the background or not running at all. The app schedules notifications with the operating system to be delivered to the user at a certain time.

**Stack –** comparable to a FileMaker database…however a Stack can contain other Stacks within it

**Card –** comparable to a FileMaker layout

**MobGUI –** LiveCode plugin (by Splash21) with custom controls, to help produce rapid mobile app development…$50 on RunRev store

**Field –** for data entry; same as in FileMaker

**Script –** performs code, creating action; same as in FileMaker

**Variables (local or global) –** usually tVariable or gVariable; same as in FileMaker

**There are four main Notification functions that we use in FM Notify –**

**mobileCreateLocalNotification, localNotificationReceived, mobileCancelLocalNotification, and mobileCancelAllLocalNotifications.**

**The following are their entries from the LiveCode Dictionary:**

**mobileCreateLocalNotification**

**Type:** command

**Syntax:**

mobileCreateLocalNotification *alertBody*, *alertButtonMessage*, *alertPayload*, *alertTime*, *playSound* [, *badgeValue*]

**See Also:** [**mobileCancelLocalNotification Command**](dictionary/command/686.xml), [mobileCancelAllLocalNotifications Command](dictionary/command/685.xml), [mobileGetNotificationDetails Function](dictionary/function/1149.xml), [localNotificationReceived Message](dictionary/message/1666.xml)

**Introduced:** 5.5

**Platforms:** Mobile

**Supported Operating Systems:**

**Summary:**

Schedules a new Local Notification with the operating system.

**Examples:**

mobileCreateLocalNotification "The Alert Body", "Show Me", "It is time to do some Exercise", the seconds + 10, true, "1"

\_

Use the **mobileCreateLocalNotification** [command](glossary/c/99.xml) to schedule a Local Notification with the operating system from within a [stack on iOS or Android](dictionary/object/1811.xml).

\_\_

**Parameters:**

*alertBody* - The text that is to be displayed on the notification dialog that is raised when the application is not running.

*alertButtonMessage* - The button text on the notification dialog that is to appear on the button that launches the application, when the application is not running.

*alertPayload* - A text payload that can be sent with the notification request. This payload is presented to the user via the [localNotificationReceived](dictionary/message/1666.xml) message.

*alertTime* - The time at which the alert is to be sent to the application. This parameter needs to be set in seconds and is the number of seconds since the UNIX Epoch, at which the notification should be triggered.

*playSound* - A boolean to indicate if a sound is to be played when the alert is received.

*badgeValue* - The number value to which the badge of the application logo is to be set. 0 hides the badge. >0 displays the value on the badge.

\_\_ \_

**Value:**

The id of the local notification that is created is returned by the [result](dictionary/function/1210.xml) function.

\_

**Comments:**

This command schedules Local Notifications with the iOS or Android operating sytem and delivers the *alertPayload* to the application. The *alertPayload* is delivered using the [localNotificationReceived](dictionary/message/1666.xml) message.

\_

**localNotificationReceived**

**Type:** message

**Syntax:**

localNotificationReceived *message*

**Objects (or Types):** stack

**See Also:** [**iphoneGetNotificationBadgeValue Function**](dictionary/function/1089.xml), [mobileCreateLocalNotification Command](dictionary/command/695.xml), [mobileCancelLocalNotification Command](dictionary/command/686.xml), [mobileCancelAllLocalNotifications Command](dictionary/command/685.xml), [iphoneSetNotificationBadgeValue Command](dictionary/command/648.xml)

**Introduced:** 5.5

**Platforms:** Mobile

**Supported Operating Systems:**

**Summary:**

Sent when the application receives a local notification from the operating system.

**Examples:**

on localNotificationReceived tMessage

 answer "Local Notification:" && quote & tMessage & quote with "Okay"

end localNotificationReceived

\_

Handles the **localNotificationReceived** message to perform an action when the operating system schedules a local message for the application.

\_\_

**Parameters:**

*message* - The Notification Message received. This is text that was submitted when the message was scheduled with the operating system.

\_\_ \_ \_

**Comments:**

The **localNotifiationReceived** message is handled once the application receives a local notification from the operating system.

Depending on the status of the application, iOS can launch a dialog box or open the login screen to inform the user that an application has information for them. Android can create an application icon in the status bar. The user can then decide whether or not to open the application. If the application is opened as a result of the notification, then LiveCode can handle the message. If an application is currently running when it receives a notification, then LiveCode can handle the message and process it as required.

If a badge number is received with the notification, then that badge value is displayed on the applications icon (iOS only), but only if the application is not currently running. If the application is running, then a badge value update does not take place, as the application can handle the message without being launched.

\_

**mobileCancelLocalNotification**

**Type:** command

**Syntax:**

mobileCancelLocalNotification *notification*

**See Also:** [**mobileCancelAllLocalNotifications Command**](dictionary/command/685.xml), [mobileCreateLocalNotification Command](dictionary/command/695.xml), [mobileGetNotificationDetails Function](dictionary/function/1149.xml), [mobileGetRegisteredNotifications Function](dictionary/function/1150.xml), [localNotificationReceived Message](dictionary/message/1666.xml)

**Introduced:** 5.5

**Platforms:** Mobile

**Supported Operating Systems:**

**Summary:**

Cancel a particular scheduled Local Notification.

**Examples:**

mobileCancelLocalNotification "3"

\_

Use the **mobileCancelLocalNotification** [command](glossary/c/99.xml) to cancel a particular Local Notification with the operating system from within a [stack on iOS or Android](dictionary/object/1811.xml).

\_\_ \_ \_

**Comments:**

This command cancels a Local Notification with the iOS or Android operating sytem.

\_

**mobileCancelAllLocalNotifications**

**Type:** command

**Syntax:**

mobileCancelAllLocalNotifications

**See Also:** [**mobileCancelLocalNotification Command**](dictionary/command/686.xml), [mobileCreateLocalNotification Command](dictionary/command/695.xml), [mobileGetNotificationDetails Function](dictionary/function/1149.xml), [mobileGetRegisteredNotifications Function](dictionary/function/1150.xml), [localNotificationReceived Message](dictionary/message/1666.xml)

**Introduced:** 5.5

**Platforms:** Mobile

**Supported Operating Systems:**

**Summary:**

Cancel all scheduled Local Notifications.

**Examples:**

mobileCancelAllLocalNotifications

\_

Use the **mobileCancelAllLocalNotifications** [command](glossary/c/99.xml) to cancel all Local Notification with the operating system from within a [stack on iOS or Android](dictionary/object/1811.xml).

\_\_ \_ \_

**Comments:**

This command cancels all Local Notifications with the iOS operating sytem.

\_