Blog

LiveCode Lessons (copy)

Enter a keyword or topic to search **Q** Search

do I use stacks with LiveCode Server?

Services

Topics

- **Installing LiveCode Server** 6
- **Using LiveCode Server** 4

How do I add Multiple LiveCode Files in LiveCode Server?

How do I use stacks with LiveCode Server? (Current Article)

How to display errors when using LiveCode Server

Sending Emails From LiveCode Server Scripts

Interacting with LiveCode Server 6

Last Updated

Apr 04, 2016

Print Article

Other Resources

Getting Started with LiveCode

Get Up and Running with LiveCode Getting Started with LiveCode Development

LiveCode Lessons

How To - Step-By-Step Guides To Tasks In LiveCode

How To - LiveCode Server Tasks

How To - LiveCode Mobile Tasks

How To - LiveCode Sample Scripts How to - LiveCode Marketplace Products

How to Purchase and License LiveCode

Tutorials

Creating a Video Library Application

Data Grid

LiveCode Data Grid Data Grid Tips & Tricks Converting the Stock Program

LiveCode Releases

LiveCode 6.5 LiveCode 6.7 LiveCode 8

Comments

Liquid error: internal for this article

How do I use stacks with LiveCode Server?

The LiveCode Server product brings our english like language to the server environment. The server engine is a separate build of the LiveCode engine with specific syntax and functionality that makes it suitable for use in command-line contexts and, in particular, as a CGI processor.

LiveCode Lessons (copy) / LiveCode Lessons / How To - LiveCode Server Tasks / Using LiveCode Server / How

This lesson will teach you how you can use stacks with LiveCode Server.

The LiveCode Server message path

The LiveCode Server engine supports loading stacks in a similar manner to that of the desktop engines. Stacks provide the programmer with an additional means to store data and manage code, allowing for improved scope and funciton name management. However, visual and graphical commands are not supported (e.g. export snapshot).

At start up, a 'Home' stack is created which serves as the container for the global script. This stack sits at the root of the message path and works in much the same way as the IDE home stack - it sits after all mainstacks in the message path, but before library stacks and backscripts. Global script execution begins with the main script file.

Using library stacks

To load a library stack use the start using command:

start using stack "<path to stack>"

As with the start using command in other environments, the libraryStack message will be sent to the newly loaded library stack and its the stack script of the will sit behind the Home stack.

Using go stack

The go stack command is also valid:

go stack "<path to stack>"

Here, the newly loaded stack will sit in front of the home stack and will be sent the standard initialisation messages (preopenstack, openstack etc).

Sending messages to stacks

You can also send messages directly to stacks using the following form:

send "<message name>" to stack "<path to stack>"

This will result in the handler <message name> of stack defined in <path to stack> being called.

Prev: How do I add Multiple LiveCode Files in LiveCode Server?

Next: How to display errors when using LiveCode Server

Comments

Robert Mann Monday Aug 01 2011 at 08:08 PM

The command "go" does not work on on-rev as of today 1st august 2011. Instead use "set the defaultStack to.."

=> This is important in order to restore the right path for all handlers and functions that will be called within the stack. Otherwise, handlers will be called in the context of the "home" stack that collect all "flat" handlers issued from .lc files.

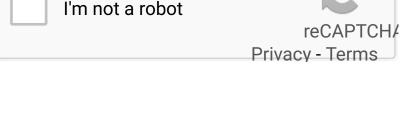
Email*

Add your comment Name*

Comment*

Subscribe E-Mail me when someone replies to this comment

Are you human?



Submit Comment