

LIVECODE (<https://livecode.com>)Platform (<https://livecode.com/products/livecode-platform/>)Resources (<https://livecode.com/resources/>)Pricing (<https://livecode.com/products/livecode-platform/pricing/>)Services (<https://livecode.com/services/>)Blog (<https://livecode.com/blog/>)login (<https://livecode.com/login/>)

Dictionary

Guides

Lessons

Courses

(<https://livecode.com>) (<https://livecode.com/resources/>) (<https://livecode.com/products/livecode-platform/>) (<https://livecode.com/pricing/>) (<https://livecode.com/services/>) (<https://livecode.com/blog/>) (<https://livecode.com/login/>) (<https://livecode.com/products/learn/>)

Files and Folders Part 1

A very common task when writing a program is to get a list of files or folders and either display it for the user to choose one, or go through each file or folder and perform some action. This lesson shows some simple methods of listing and sorting files and gives some hints on the relevant LiveCode commands used.

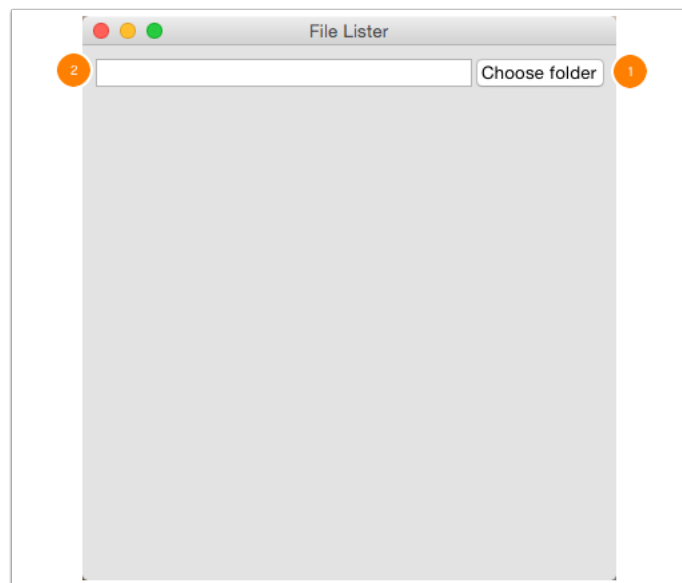
Please download a copy of the supporting stack `File Lister.rev` for a working example of the code that follows

Getting a list of files or folders

LiveCode has a very simple way of obtaining lists of files and folders on a computer. The two functions **files** and **folders** and the property **defaultFolder** are the only things you need. The basic idea is to set the **defaultFolder** to the folder you want to work with, and then use the two above-mentioned functions to return the lists of files or folders.

Choosing a folder

We will create a stack that allows us to select the folder we want to work with and displays the files and folders in that folder. Firstly add a field and a button to the new stack, which we will name `File Lister`.

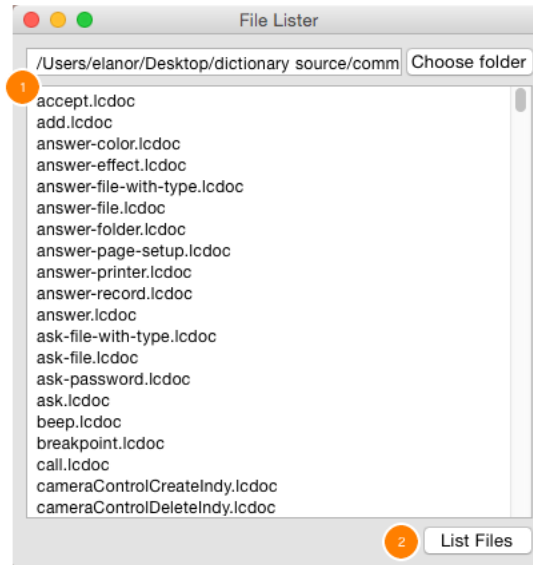


Then we add some code to our "Choose folder" button(1), using the **answer folder** command to select the folder we want to work with. This puts the full path to the selected folder into the field we have named "Folder"(2)

```
on mouseUp
    local tFolder

    answer folder "Please choose the folder you want to work with"
    put it into tFolder
    if there is a folder tFolder then
        put tFolder into field "Folder"
    end if
end mouseUp
```

Displaying a list of files



Now we will add a scrolling field called "List"(1) and a "List files" button(2) to our stack.

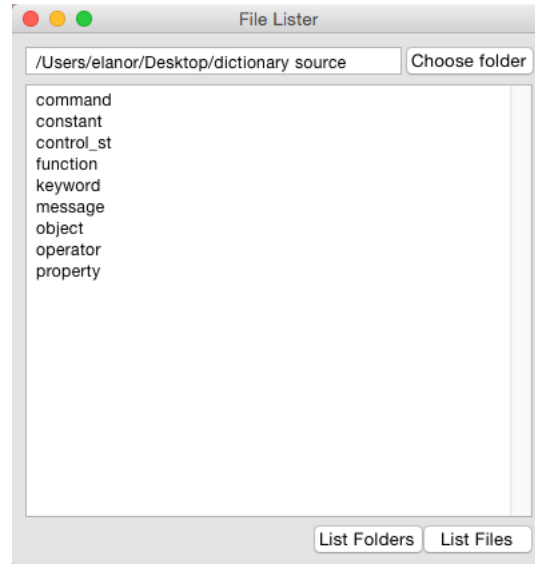
We then use the **files** function to get a list of the files in our selected folder and display the list in the field

```
on mouseUp
    local tFolder

    put field "Folder" into tFolder
    set the defaultFolder to tFolder

    put the files into field "List"
end mouseUp
```

Displaying a list of folders



You can use the folders function in exactly the same way, just add another button and change the last line of the mouseUp handler to

```
put the folders into field "list"
```

Filtering the lists

On many file systems the folders "." and ".." refer to the current folder and the parent folder respectively. It is important to make sure that these two values do not appear in the list of folders or files because they are not real folders and more importantly as you can see later, they can cause problems in your program. A good way to safeguard yourself from this is to use the following functions instead of using the files or the folders. We add these functions to our stack script and change our buttons to call these functions.

```

# Filters the strings "." and ".." from a list
function filterDots pList
    local tList

    put pList into tList
    filter tList without "."
    filter tList without ".."

    return tList
end filterDots

# Returns a filtered list of files in the current directory
function filteredFiles
    return filterDots(the files)
end filteredFiles

# Returns a filtered list of folders in the current directory
function filteredFolders
    return filterDots(the folders)
end filteredFolders

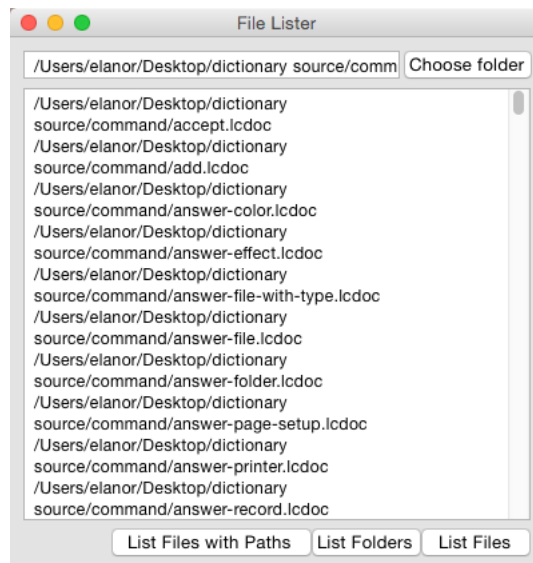
## List Files button script
on mouseUp
    local tFolder, tFileList

    put field "Folder" into tFolder
    set the defaultFolder to tFolder

    put filteredFiles() into field "List"
end mouseUp

```

Listing the full paths to the files



Sometimes it will be useful to know the full path to a file, lets add another function to our stack script which will display the full path to each file in the "List" field. Again we will add a button to the stack that calls the handler on the card script.

```

function filteredFilesWithPaths
  local tFiles, tFilesWithPaths

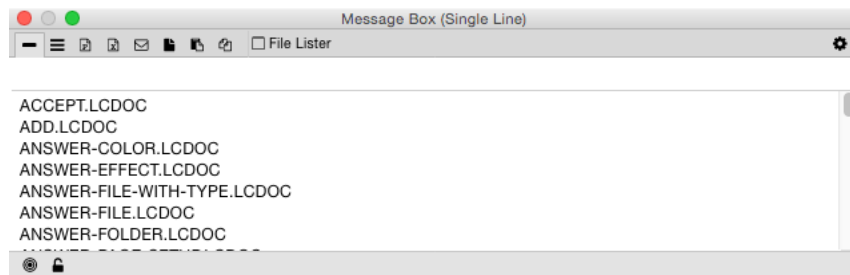
  put filteredFiles() into tFiles

  ## Use a repeat loop
  ## For each file we put the defaultFolder before the file name, this gives the full path to the file
  ## We use this to build a new list which we return to be displayed
  repeat for each line tFile in tFiles
    put the defaultFolder & slash & tFile & return after tFilesWithPaths
  end repeat
  delete the last char of tFilesWithPaths

  return tFilesWithPaths
end filteredFilesWithPaths

```

Performing an action for each file



Performing an action for each file in a folder is a common task. This handler will loop through each file in a folder and call the doSomething handler for each file, giving the handler the name of the file.

All that is left to do is write the doSomething handler. A simple action could be to make the name of the file upper case and write it to the message box for example:

```

# Use this template function to perform an action on each file in a folder
on doForEachFile pFolder
  set the defaultFolder to pFolder

  ## Use the filtered list of files
  repeat for each line tFile in filteredFiles()
    ## Call the doSomething handler for each file
    doSomething (tFile)
  end repeat
end doForEachFile

on doSomething pFile
  ## As an example we will just make the filename upper case
  ## and print it to the message box
  put toUpper(pFile) & return after msg
end doSomething

```

Next lesson

In the next lesson we will look at using recursion to perform similar tasks to those above but including subfolders and their files.

Files and Folders Part 2 (<http://lessons.livecode.com/m/4071/l/17080-files-and-folders-part-2>)

13 Comments

Marilyn Thursday Mar 01 2012 at 05:22 PM

I have created a stack that includes two text files. The stack works perfectly on my computer. When I deploy into a "stand alone" application it works perfectly on my computer. But when I put the stand alone application onto my husband's computer, the text files are missing.

When I share the LiveCode app with another computer, do I have to copy the whole folder? Surely the stand alone LiveCode app should have the files contained within it?

When deploying for sharing, or sale, people only download the LiveCode file.

I would appreciate if you could help me with this as I have been struggling with it for weeks.

Note: I am currently deploying to Windows, and hope to deploy to Android later.

Thank you,
Marilyn

Elanor Buchanan Friday Mar 02 2012 at 11:34 AM

Hi Marilyn

With a Windows standalone the .exe file does not include any supporting files, you need to distribute these with the executable either by zipping up the whole folder or by creating an installer.

When you get to deploying to Android the supporting files will be included in the .apk file created by the standalone builder so you won't need to worry about it for Android.

I hope that helps.

Elanor

Marilyn Sunday Mar 04 2012 at 09:45 PM

Elanor,

That does answer my questions as to what to expect from LiveCode.

Thank you,
Marilyn

Marilyn Saturday Apr 07 2012 at 09:21 PM

Elanor,

I am trying to do something very simple. I have a text file called "Answers". What I would like to do is put this file into a variable so that I can access the data from anywhere in my stack.

This works:

```
on preOpenCard
put specialfolderpath("desktop") & "/Mat Res/Text Files/Answers.txt" into tFilePath
put URL ("file:" & tFilePath) into tAnswers
```

```
put tAnswers into field "Field"
end preOpenCard
```

However, when I split them up :

```
preOpenCard
put specialfolderpath("desktop") & "/Mat Res/Text Files/Answers.txt" into tFilePath
put URL ("file:" & tFilePath) into tAnswers
end preOpenCard
```

Then try to use a button to:

```
on mouseUp
put tAnswers into field "Field"
end mouseUp
```

I just get tAnswers in my field instead of the contents of the variable tAnswers.

Could you tell me how I can set up the variable on preOpenCard, so that I can access it from anywhere in my stack?

Thank you,
Marilyn

Hanson Schmidt-Cornelius Tuesday Apr 10 2012 at 12:03 PM

Hi Marilyn,

this has to do with the scope of the variable "tAnswers". You need to declare where the variable "tAnswers" is to be used. Have a look at the entries for commands "local" and "global" in the LiveCode dictionary.

Kind Regards,

Hanson

Liz Friday Jun 29 2012 at 06:55 PM

I'm trying to filter the spaces out of a file, so I don't have to worry about whitespace affecting the data. Here's the code I currently have on the card:

```
function filterSpaces spacedText
put spacedText into workText
filter workText without " "
//filter workText without ""
```

Hanson Schmidt-Cornelius Tuesday Jul 10 2012 at 01:48 PM

Hi Liz,

I am not exactly sure what you are trying to achieve here. More information about the kind of data you are processing would be useful. Are you only interested in single spaces or do you have to take account of multiple spaces, tabs and possibly returns?

"filter" can be used to remove lines that match a particular pattern. If you are trying to remove white space from a text buffer in memory then also have a look at "replace" or "replaceText".

Kind Regards,

Hanson

Marcus Wednesday May 29 2013 at 08:03 PM

Hi

I am trying to write a small app to move files into separate folders according to their extensions, to clean up my download folder once a day. I have found revCopyFile as well as revMoveFolder but cannot find a move file command. I do not want to make copies of my files, but just move the files to individual folders.

Any ideas?

kind regards and thanks
Marcus Baker

Hanson Schmidt-Cornelius Friday May 31 2013 at 06:23 PM

Hi Marcus,

you can use the "rename" command to move files from one directory into another.

Kind Regards,

Hanson

Marcus Baker Friday May 31 2013 at 06:58 PM

Dear Hanson

Thank you so much for your help, the rename command works great :=)

Do you know if anybody ever created a list of uses and which commands cover them? The dictionary lists a command and what it does, but I have not found a list that reverses that, so that you can search by what you need and it will list the command for it.

When I search for move or file, the dictionary does not list the rename command, but when I know of the rename command and search for it, the dictionary lists it for me with all its uses.

kind regards
Marcus

Hanson Schmidt-Cornelius Monday Jun 03 2013 at 01:47 PM

Hi Marcus,

great you got it working.

With regards to your suggestion of creating a list of references to a particular functionality in the database. It would be great to have this and we have thought about this and even considered how such a relational system would operate, but unfortunately we have not yet scheduled time to implement this kind of feature. So, even though it would be great to make this available, I cannot give you any idea if or when it would be implemented.

Kind Regards,

Hanson

Nick Tuesday Sep 17 2013 at 06:59 PM

How wuld you then go about sorting the list by date created or similar?

Ideally I'd like to display the latest file created in a folder

Hanson Schmidt-Cornelius Friday Sep 27 2013 at 02:13 PM

Hi Nick,

as a start, try using "put the detailed folders into field", rather than "put the folders into field". You can then filter and sort on the additional information that is provided.

Have a look at "sort" in the LiveCode dictionary.

Kind Regards,

Hanson