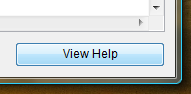
WordLib

*The Word Processing Import Library for LiveCode*

**Version 2.3 Quick Reference**

Viewing this document

Within LiveCode: **Try WordLib** stack, **View Help** button.



Externally: **PDF** and **MS Word** formats included in the WordLib folder.

Examples/Help/Etc.

Open the Try WordLib stack in LiveCode. That’s the starting point!

Loading the library

start using "wordlib.rev" -- (located in defaultFolder)

start using "[path/]wordlib.rev" -- (located in a folder)

start using (specialfolderpath("resources") & "/wordlib.rev")

-- (same folder as your project stack)

start using "wordlib" – (embedded as a substack)

Supported file formats

MS Word 2007-2020+ (.docx .docm .xml) Extensive support

OpenOffice (.odt) Strong support

MS Word 2003 (.xml) Strong support

MS Word 97, etc. (.doc) Limited support

Importing files

General use:

importDocumentToField fieldName [,promptText] [,startFolder]

-- imports any supported document to field fieldName

-- (includes file dialog and error checking)

-- (startFolder can be folder path or "#/path" relative to stack)

Note: You can now use importDocumentToField for Mobile apps!)

set htmltext of field fieldName to htmltextOfDocument(filePath)

-- import any supported document

Specific formats:

set htmltext of field fieldName to htmltextOfDocX(filePath)

-- .docx .docm .xml

set htmltext of field fieldName to htmltextOfDoc(filePath)

-- .doc

set htmltext of field fieldName to htmltextOfODT(filePath)

-- .odt

Meta information

global gwordlibMetaInfo -- array of properties from document

get gwordlibMetaInfo["Lines"]

-- MS Word's properties: ["Creator"], ["Words"], etc.

-- plus human-readable Word version: ["WordProcessor"]

global gwordlibLoadTime -- last import time in seconds

-- (time to produce htmlText; LiveCode field rendering not included)

global gwordlibLastFilePath -- file path of most recent import

set htmltext of field fieldName to htmltextOfDocument(gwordlibLastFilePath)

-- reload a file, for example after external changes

Error checking

Basic check:

get htmltextOfDocument(filePath)

if item 1 of it is "error" then

-- major error, document will not be imported

put item 2 of it into errorCode

put item 3 to -1 of it into errorMessage

answer errorMessage

else

-- import successful

set htmltext of field fieldName to it

end if

Minor errors (advanced use, debugging):

global gwordlibErrors

if gwordlibErrors is not "" then

-- minor errors and warnings

repeat for each line err in gwordlibErrors

put item 1 of err into errorCode

put item 2 to -1 of err into errorMessage

end repeat

end if

File types:

get wordlibFileTypes() -- currently "docx docm xml odt doc"

get wordlibSupports(fileName) -- whether document is supported

Bundled documents

A) Import a file located relative to current stack/standalone:

importBundledDocument fileName, fieldName [,"silent"]

-- (silent option to suppress default error dialog)

put wordlibBundled(fileName) into theFile

set htmltext of field fieldName to htmltextOfDocument(theFile)

(For Mobile, use Copy Files in Standalone Settings to specify which documents to bundle!)

B) Store an imported document permanently inside a LiveCode stack:

saveBundledDocument fieldName, stackProperty

-- save field to custom property, import associated images

set htmltext of field fieldName to stackProperty

-- display a document saved with the previous command

deleteBundledDocument stackProperty [,"silent"]

-- delete stored document and associated images

Image management

Media folder:

global gwordlibMediaFolder -- temp folder for images

global gwordlibAutoCleanup -- delete temp files on startup if true

openWordLibMediaFolder -- open Media Folder in desktop

cleanWordLibMediaFolder -- delete all stored image/media files

Displaying unsupported images in documents:

global gwordlibImageLinks

-- show icon and link for unsupported images, if true

global gwordlibImageLinkIcon

-- icon to use for display of unsupported images

Comments

By default, comments are linked: "Comment: #ID 'TEXT' (AUTHOR, SYSTEM DATE TIME)"

get linkText of word 24 of field "doc"   
--> "Comment: #0 'LOL!' (Curry, 4/25/2022 7:09 AM)"

get metaData of word 24 of field "doc" --> "CommentIDs:2,5"

global gwordlib -- new array for all future WordLib user options

put false into gwordlib["LinkComments"] -- add linkText? (boolean)

put true into gwordlib["HiliteComments"] -- use backColor?

put "yellow" into gwordlib["HiliteComments"] -- (boolean or color)

global gwordlibComments[2]["text"] -- text of comment ID 2

global gwordlibComments[7]["author"] -- author of comment ID 7

global gwordlibComments[1]["date"] -- abbrev English date && time

ccFieldCommentRanges(pHtmlText) --> "ID Start,End Excerpt" per line

Advanced options

global gwordlibDoubleCR -- force double line breaks, if true

Experimental (imperfect, but may be useful):

global gwordlibFieldHacks –- workaround LC features, if true

global gwordlibTextZoom – resizes text during import, % as integer

Deprecated (no longer relevant):

global gwordlibUnicode -- import in Unicode (true), ANSI (false),

-- or smart mode using Unicode where necessary ("auto", default)

-- This was an awesome feature for LC 4.x; no longer needed!

Library information

get wordlibVersion() -- library version as five-digit integer

get wordlibVersion(".") -- library version in 1.2.3.4 format

get wordlibInfo() -- library name and dotted version - author

Update/Registration

**WordLib** web page:

<http://livecodeaddons.com/wordlib.html>

Secondary backup web page (less current):

<http://www.curryk.com/wordlib.html>

**Purchase** button in the **Try WordLib** stack:

Graphical user interface, text, application

Description automatically generated

**License key**:

on preOpenStack

start using "wordlib.rev"

registerWordLib "YOUR-KEY-HERE"

end preOpenStack

**TOS**: Password-protect distributed stacks to keep your key private!

**Standalones**: Include wordlib.rev, cclib.rev, RevZip.