BYU | BRIGHAM YOUNG UNIVERSITY

Office of Digital Humanities

Back BYU LiveCode Lessons Gateway

Mac Creator Signature and File Types in Revolution/LiveCode (For use with the fileType and stackFileType properties)

by Devin Asay

Creator Signatures and File Types

The Mac Classic operating system used four-character codes to keep track of what application should be used to open specific files and what type of files applications could open. A unique creator signature was assigned to every application and any file could store a creator code in its resource fork. The OS maintained a database of which creator codes matched which applications, so that when the user double-clicked a file, the OS would look up the creator code in its database and launch the proper application. Similarly, every file could have a file type code assigned to it. For example, plain text files were type 'TEXT' and MS Word documents were 'WDBN'. Application writers could tell their applications which types of files it could open. So many different applications could open TEXT files, but very few could open WDBN files.

When Mac OS X came along it had the ability to determine appropriate applications for launching files by looking at the file's extension, as had been done in the DOS, UNIX and Windows world for years. In addition, Mac OS X can still recognize and match files to applications using file types and creator signatures.

Note: Beginning with OS X 10.6 Snow Leopard, creator codes are ignored by OS X, and have been superseded internally by Apple's Uniform Type Identifier scheme. However, setting a filetype in LiveCode can still be useful for providing support for OS X legacy systems. In any event, no harm is done by setting this property in your LiveCode projects.

When writing files to the local file system with LiveCode on Mac systems, it is a good idea to first set the fileType global property, so that the OS will know what applications can open it. The syntax is:

```
set the fileType to creator & type
```

Creator & type is an eight-character string composed of a four-character creator signature followed by a four-character file type. So for example, to save a PDF file assigned to be opened in Adobe Reader use

```
set the fileType to "CAROPDF "
```

To save a PDF file to be opened in the default PDF application set in the system's preferences, use the creator signature for an unknown application, "????":

```
set the fileType to "????PDF "
```

To save a movie file to be opened in QuickTime Player

```
set the fileType to "QVODMoov"
```

If you set the fileType to empty, OS X will look at the file's extension and determine which application should open it based on the system's preferences.

Note: If you want to save a stack with a creator and file type that are different from the LiveCode default "RevoRSTK", set the **`stackFileType`** property instead.

See the LiveCode Dictionary entries for 'fileType' and 'stackFileType' for more information.

Discovering Creator and File Type Codes

The difficulty with setting these codes is that there is no single, authoritative list available to the public. So people have resorted to maintaining lists on their own. Here is a list of commonly-used creator signature and file type codes, culled from my own experience and lists I discovered on the web:

Creator Signatures

Adobe Reader	'CARO'
iTunes	'hook'
Any Application	'APPL'
SimpleText (Classic App)	'ttxt'
TextEdit (OS X App)	'ttxt'
Unknown application	'????'
LiveCode	'Revo'
BBEdit	'R*ch'
TextWrangler	'!Rch'
TextMate	'TxMt'
Photoshop	'8BIM'
GraphicConverter	'GKON'

File type codes

text file	'TEXT'
PDF	'PDF '
QuickTime movie	'MooV'
MPEG video	'MPG '
MPEG2 movie	'MPG2'
MPEG-4 video	'M4V '
MP3 audio file	'Mp3 '
LiveCode stack	'RSTK'
HyperCard stack	'STAK'
GIF file	'GIFf'
PNG file	'PNGf'
JPEG file	'JPEG'
BMP file	'BMPf'

QuarkXpress	'XPR3'	TIFF image	'TIFF'
WordPerfect Mac	'WPC2'	EPS file	'EPSF'
MS Excel	'XCEL'	Photoshop PSD	'8BPS'
MS Word	'MSWD'	Mac PICT image	'PICT'
Stuffit Expander	'SITx'	MIDI music file	'Midi
InDesign	'InDn'	QuarkXpress document	'XDOC'
Preview	'prvw'	WordPerfect Mac	'WPD1'
Safari	'sfri'	MS Excel worksheet	'XLS '
Firefox	'MOZB'	MS Word	'WDBN'
Garage Band	'band'	MS PowerPoint	'PPT3'
Keynote	'keyn'	Stuffit archive	'SIT!'
Pages	'page'	RTF files	'RTF '
Numbers	'NMBR'	Mac snd resource	'snd '
iWeb	'iweb'	Windows WAV file	'WAVE'
Dreamweaver	'DmWr'	AIFF file	'AIFF'
NeoOffice	'NO%F'	µ-law sound file	'ULAW'
Real Player	'PNst'	Generic binary file	'BINA'
DiskImageMounter	'ddsk'	TrueType font	'tfil'
		Finder	'FNDR'
		Flash file	'SWFL'
		Universal Disk Image (DMG)	'udif'

If you can't find the creator code you need in this list, you can get creator codes from the Info.plist in a Mac OS X application bundle. Look at the value for node CFBundleSignature.

Here are some URLs to sites where people who have tried to construct lists of common creator and file types. Since the web is constantly changing, these sites may disappear at any time.

http://mail.python.org/pipermail/pythonmac-sig/2005-February/013028.html
http://www.zeusprod.com/technote/filetype.html
http://kb.iu.edu/data/aemh.html

Making Up Your Own Creator Codes

If you create your own custom applications you can still make up your own, custom creator signature. The requirements are:

- It must consist of four characters. A space is a legal character. Upper and lower case alpha characters are considered different characters.
- It cannot consist of all lower case alpha characters, which are reserved for Apple applications.
- The combination of characters must be unique.
- You should register your creator code with Apple at http://developer.apple.com/datatype. This ensures that your code does not conflict with any other registered application.

Once you have made up and registered your creator code, use it in the Signature field in the OS X settings in the LiveCode Standalone Applications Settings utility.

Making Up Your Own File Types

Apple no longer registers file type codes, but you can still create and use your own if it seems useful to do so. The rules for file type codes are roughly the same as those for creator codes. Since there is no way to register them to ensure they are unique, the best approach is to try to make up a file type that you have never encountered before. The lists included here and in the URLs above can help in this. But even if you choose to use a file type code that already exists, when you use it in conjunction with a creator code, the combination is still unique. For example, if you use LiveCode stacks as documents that your Rev application produces, you should be fine to use the 'RSTK' file type with it. Let's say your application uses a custom creator code 'FooB'. If your application creates stacks as document files for its own use, just set the stackFileType like this before you save it to disk:

```
set the stackFileType to "FooBRSTK"
```

That will tell the OS who owns the document file and should prevent confusion should the user try to open a document file belonging to your application by double clicking it.

Using a Custom File Extension

A final caveat: if you use a custom file type and creator code, you'll be best off to also use a custom file extension. Using the previous example, if your application saves LiveCode stacks as custom documents, you should also use your own custom file extension. A document file that has the stackFileType "FooBRSTK" but a .rev file extension may still be considered by the OS to belong to the LiveCode application, not yours. This is especially true in the Windows and Linux environments, since creator codes and file types are only used in Mac OS Classic and OS X. It is better to save your stack-as-custom-document with its own extension, like this:

```
set the stackFileType to "FooBRSTK"
save stack "MyDoc" as "/Folder/MyDoc.foob"
```

Back BYU LiveCode Lessons Gateway

Maintained by Devin Asay.