

### Disclaimer

All information in this document is Copyright © 2023 Dell Krauchi, Sentinel Music Studios and RML Labs. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage or retrieval system, or otherwise, without prior permission and written consent of Dell Krauchi, Sentinel Music Studios and RML Labs. All trademarks and brand identities as used in this text are used for descriptive purposes only and remain the property of their respective owners.

### Preamble

According to the User Manual, the visual aspects of RML labs software consists of "...*graphic libraries...which completely control how the user interface looks.*"

These graphic libraries are stored in ".dll's – or "dynamic link libraries" with these .dll's further consisting of "bitmap images" which are referred to as "resources".

The total number of these image resources within these .dll's range from 2 to 261...as shown in the table below:

No.	Name of .dll	Bitmaps
1a	Studio_FX_Eq.dll	2
1b	Studio_FX_Eq64.dll	2
2a	Studio_FX_Echo.dll	3
2b	Studio_FX_Echo64.dll	3
3a	Studio_FX_FreqAnalyzer.dll	4
3b	Studio_FreqAnalyzer64.dll	4
4a	Studio_FX_Levelizer.dll	7
4b	Studio_Levelizer64.dll	7
5	Studio_FX_Reverb.dll	13
6a	SAWStudioFx_VideoTrackViewer.dll	34
6b	Studio_FX_VideoTrackViewer64.dll	34
7a	Studio_Midi_WorkShop_Graphics.dll	45
7b	Studio_Midi_WorkShop64_Graphics.dll	45
8a	SAC_Graphics.dll	257
8b	SAC64_Graphics.dll	257
9a	SAWStudioGraphics.dll	261
9b	SAWStudio64Graphics.dll	261

**Table 1**

- Note: The use of the above colour scheme is simply to note those .dll's *that are common*. More on this usage later.

### Controlling "the Look" of the User Interface

If the "look" of a particular user interface of any of the above elements just happens to *not be to your liking*, then there are *two options available to you* – first, you could *update the existing images yourself*, or second, you could employ images *that were developed by others*. This text – for the most part, will focus *entirely on the latter*.

## Employing Existing Images

These existing images are generally made available *in one of three methods*, 1) as "*shades*", 2) as ".*dll's*", or 3) as "*bitmaps*". The actual employment of these three methods is as follows:

### As "*shades*"

Shades are available only for use within SAC and SAW – "*item 8*" and "*item 9*" above, and are deployed via the File > Shades menu option. The file extension for such data is ".*shd*".

### As ".*dll's*"

The use of .*dll's* are generally reserved for use with the native effects – "*item 1*" to "*item 7*" above, though it is possible to employ the use of .*dll's* with SAC and SAW as well.

- Observation: The latter is the method employed when purchasing the SAC and SAW software.

However, due to certain "*legality issues*", the updating of the native effects .*dll's* is deployed via the use of "*bitmaps*".

- Observation: This latter method is deployed through the employment of a "*resource application*" – such as "*Resource Hacker*". More on the use of this application later.

### As "*bitmaps*"

The use of such bitmaps is generally reserved for "*developers*" – whom, upon completion, will provide either "*shades*" – which are reserved for SAC and SAW only, and "*bitmaps*" – for employment with the native effects and the subsequent use of the Resource Hacker application.

### The "*Legal*" Things

As a direct result of the specific native effects .*dll's* being "*commercial offerings*" – simply meaning that such items *must first be purchased*, as is the case with "*item 1*" to "*item 7*" above, developers make provision for the updating of such .*dll's* through the use of *bitmaps only*.

### Commercial Offerings

Third-party developers also have the option of 1) charging for their development efforts – thus making such offerings "*commercial in nature*", and which subsequently, would require the purchase of such offerings, or, 2) not charging for their development efforts- and whom therefore, have decided *not to charge for their development efforts*, and are thusly, making the .*shd* and bitmaps available "*for free*".

### Disclaimer

*No commercial offerings whatsoever have been employed in this text and the subsequent procedures. However, the following procedures will function as expected for any-and-all commercial products employed as well<sup>1</sup>.*

### The Data Sources

The bitmaps that are being employed here in this present text *are derived from three sources* namely, 1) the 32-bit/64-bit demo versions of the Echo/Delay and Eq native effects, 2) HapHazard's "*Glass\_House\_v4.7*" theme for SAC – which also includes the image resources for the Echo/Delay and the Eq native effects, and 3) from Ivan Jurasik's "*Something*" theme for SAW. *Thank you all for these image resources and the efforts in the development of these image resources! Very much appreciated!*

### About Resource Hacker

Resource Hacker is *a resource editor* for 32-bit and 64-bit applications. And while Resource Hacker is primarily a GUI-based application, Resource Hacker also provides many options for compiling and decompiling resources *from the command-line* – the latter which will be employed extensively here in the accompanying "*scripts*".

## About Re-Development

For those that have an interest in developing their own graphics – from scratch, the following information will be of some assistance to you.

### General Information

It will be helpful to understand that the total number of resources contained within "like .dll's" – is that these resources *are the same* for either the 32-bit version or for the 64-bit version of that particular .dll.

- Note: See Table 1 above.

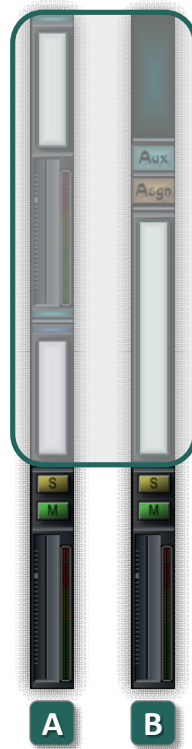
What this means then, is that the updating of a 32-bit resource to 64-bit *is entirely plausible* – and vice-versa. The only exception to this plausibility is with regards to the 32-bit and 64-bit versions of SAC.

### About SAC

Early versions of the 32-bit version of SAC contained a total of 173 image resources – with the later 32-bit versions and the 64-bit versions of SAC containing a total 257 image resources – *a difference of some 84 image resources!* With the exception of two resources, the remaining 82 resources *were specifically designed for the touch-screen functionality of SAC.*

### SAC32 versus SAC64

In particular, with regards to the two aforementioned resource exceptions, please note the following:



The above images are derived from the "F Mixer Return" channel strip – with [A] derived from the 32-bit version of SAC, and with [B] derived from the 64-bit version of SAC. In particular, please note the *obvious* differences in the upper portion of the channel strip. Therefore, if converting from the 32-bit of SAC to the 64-bit of SAC – the above image [A], would necessitate being updated accordingly.

### The Three Procedures

With each available *on a donation basis*<sup>2</sup> are three procedures for updating<sup>3</sup> and potential re-development<sup>4</sup> of the bitmap images that are stored within the various "graphics .dll's" as provided by RML labs and subsequent third-party developers.

Each procedure contains a separate cover text, the required sample data, and the necessary command scripts – with the use of the sample data ensuring that each of the three procedures can be completed successfully. These cover texts, sample data and scripts are provided in an accompanying ZIP file.

### **The Demo Versions**

As always, it is strongly recommended that the demo versions of any-and-all RML Labs software be employed in these procedures so as to leave your existing installation intact.

### **About the Three Procedures**

Each of the three procedures addresses *a specific aspect* of the updating of, and potential re-development of, the graphics contained with the various RML Labs products. A brief explanation of each of these three procedures now follows.

#### **Procedure 1**

This procedure employs previously extracted bitmaps to create 1) a new 32-bit or 64-bit main .shd, 2) a new 32-bit or 64-bit main .dll, or 3) a new 32-bit or 64-bit native effects .dll.

- Observation: The employment of this procedure might be the choice for employment with the native effects.

#### **Procedure 2**

This procedure employs a 32-bit main .shd, main .dll or native effects .dll – which is then converted to 64-bit.

- Observation: The employment of this procedure would be relevant for those that already have a 32-bit offering and who requires updating that offering to 64-bit.

#### **Procedure 3**

This procedure simply extracts the bitmap resources from a 32-bit or 64-bit main .shd, main .dll or native effects .dll for either updating or potential re-development purposes.

- Observation: The employment of this procedure might be the choice for those who have a desire to develop or to update an existing offering.

### **The ZIP file**

The contents of the ZIP file are as follows, 1) the cover text, 2) the "*working folder*", 3) the scripts, and 4) the sample data. An explanation of each of these data now follows.

#### **#1: The cover text**

This cover text is simply a procedural outline for performing one of the three aforementioned procedures.

#### **#2: The "*working folder*"**

Depending on the procedure being deployed, the working folder may consist of the following:

1. The resource\_hacker folder: This folder will be the main working folder for the procedure.
  - Observation: This folder contains the ResourceHacker.exe application.
2. bmps folder: This folder contains any-and-all respective bitmaps which will be summarily stored in their respective sub-folders.
3. existing\_dll folder: This folder contains any-and-all respective graphics .dll's.
4. new\_dll folder: This folder contains the newly, updated graphics .dll's.
5. scripts folder: This folder contains all of the required command scripts for either updating the data or extracting the data of the respective graphics .shd's and .dll's.

### #3: The scripts

As noted above, Resource Hacker also provides many options for compiling and decompiling resources *from the command-line*. Such command-line usage is made possible via the use of "scripts" – with the added benefit of the "simplifying" the use of the Resource Hacker application.

### About Data File Renaming

To maintain a degree of consistency with the use of the scripts, the filenames scheme is as follows:

1. The main applications will adopt the following filenames: new\_sac\_32\_bit.shd, new\_sac\_64\_bit.shd, new\_saw\_32\_bit.shd and new\_saw\_64\_bit.shd.
  - Note: With the exception of the native effects, the .shd names can be updated accordingly.
2. The native effects – as a result of the inherent design characteristics, will adopt the default filenames as noted in Table 1 above.

### #4: The sample data

Depending on the procedure being deployed, the sample data may consist of any-or-all of the following:

1. Demo versions of the respective main application .dll's
2. Demo versions of the respective native effects .dll's
3. "Free" .shd data files
4. Bitmap image resources as derived from any of the "free" graphics offerings

### In closing....

That is pretty much...that! I do hope that some will find this information of some practicable use? Thank you for your time and interest in the above.

Sincerely,



Date: May 10, 2023

Updated: January 17, 2024

### References

<sup>1</sup>It is hopefully obvious, that permission from the original developer should be obtained – and subsequently granted, *prior to performing such an extraction procedure*.

- It is also hopefully obvious, that though someone may own a particular .shd or .dll data file – that the actual content of that particular .shd or .dll data file is considered "*intellectual content*" and are therefore, will remain the property of the developer of that .shd or .dll data file.

<sup>2</sup>buymeacoffee or PayPalMe

- <https://www.buymeacoffee.com/jrv5rzpv7f>
- <https://www.paypal.com/paypalme/mres335>

<sup>3</sup>Such bitmaps may be have been provided 1) by "*developers*" or, 2) such bitmaps as that which has been developed and/or updated by you.

<sup>4</sup>The total number of bitmaps *is very important* and as such, the total number of bitmaps *must equal the total number of bitmaps a noted in Table 1 above*.