

You Can Create Your Own Mac Program... continued

It doesn't take long to become loyal to the Mac; even in the midst of *groupthink*, which frequently belittles Macs as computers, "for those not literate enough with technology to use *real* computers." Mac users stand by their mark and hold true to the sense that Macs are truly something special; among the throng of the ordinary. If that is your view, your senses aren't fooling you. There is something special indeed, and it all starts with the Mac OS.

No doubt, if you're a Mac user, you know the Macintosh user interface. You know the buttons, the menus, and the dialogs. You probably know how great the hardware advantages are too— ask your friend Joe PC about adding on new peripherals! <snicker>. If you've been around the block a few times with your Mac, you've probably also learned that there is a fascinating difference between the programs you use on your Mac and their PC counterparts. What truly makes the Mac special is something called *resources*.

PC files are created, basically, as a stream of bytes; one big lump of data. Mac files can also be just a single stream of data. Text files, GIF files, and other types of documents that can cross platforms, being usable on both Macs and PC's, are some examples of such files. Mac applications, however, are almost universally created from resources. MacBinary is the term created to describe this type of file format; consisting of two forks, one called the data fork, and the other the resource fork. The data fork is the equivalent of the PC file format, a single stream of data in one big lump. The resource fork is where Mac *magic* happens.

Anything can be placed into a resource. Think of a resource as an organized piece of data, like a dresser drawer, organized in a preset fashion. Resources have names, like 'MENU' for a menu resource, or 'FONT' for a font resource. Now if you had a dresser drawer that said "SOCKS" on the front, inside you would expect to find socks. Further, an organized person might have them placed, perhaps, according to color, then size, etc.. A resource is like an organized sock drawer. Apple has a free utility just for organizing those sock drawers — called ResEdit.

GREAT, NOW HOW DO I MAKE A MAC PROGRAM?!

The experience of creating a computer program comes at different levels. My first computer program was created on an Apple II computer. It looked something like this: 10 Print "Hello World";

You Can Create Your Own Mac Program... continued

Not real big, but it sure impressed me. That program was written in BASIC, a *high level* programming language. Programming languages are termed high and low according to how close they are to the level of the computer's actual machine language. A low level language is close to the machine's language. High level languages are designed to be closer to human language. Today, many applications are created in large part by pointing and clicking, using programs like HyperCard and MacApp, or, using programs like ResEdit.

KID'S CLICKS!

"Kid's Clicks!" is a program I wrote using ResEdit and a programming language called C. Kid's Click's is primarily designed for kids, but not *just* for kids. By itself, the program will do little more than stare at you, although there is a cute picture of my daughter on the "About" screen you can look at! Kid's Clicks requires modules to come to life.

With your imagination and a little time, Kid's Clicks will become *your* program. Your Mac program will be a module for Kid's Clicks!

If you don't have a copy of Kid's Clicks, you can download it from a software library online, either from your local BBS, from internet sites, or from one of the major online services. Look for it wherever you find kid's games and other games. Try out the modules included with the Kid's Clicks program.

Note: Kid's Clicks requires System 7 and 256 colors.

YOU NEED RES-EDIT!

First thing you need to create your own program is a copy of ResEdit 2.1.1 or newer. You can download ResEdit from any major online service's forum for Mac developers, and many other places, like those mentioned before. It is free, made publicly available by Apple. Once you have a copy, you're ready to begin creating your program.

LET'S DO IT!

You've got a copy of ResEdit, now you're ready to go! Ok, here's where we start: Kid's Clicks uses modules. Make a copy of a Kid's Clicks module. Remember: Copy the module, not the Kid's Clicks program! Now, launch ResEdit. Use the **Open** command under the **File** menu to open the resource fork of your copy of the module.

You Can Create Your Own Mac Program... continued

If you didn't know about ResEdit until now, this is your first look inside a Mac program! Remember, this is a copy, so even if you screw something up, don't worry, you'll just make another copy. Continually saving your module as you progress will also protect you if you make a mistake. You can always revert to the last copy saved. If you make a mistake, first try the **Undo** command from the **Edit** menu, if that doesn't work, close the module without saving it, and reload the last copy saved.

ALWAYS WORK ON A COPY, NEVER AN ORIGINAL!

What you should see after opening the module is a list of resources. You can view them as a list of names, or view them as icons. Most prefer to view the resources by their icons. Browse around. Double click on an icon or name of a resource type. This will bring up a list of the resources within the module of that resource type. Check out the 'DLOG' resources! Peek around, see what's there! If you start thinking, "Gee, I'd really like to know more about this stuff," you can get a great book called, ResEdit Complete , by Peter Alley and Carolyn Strange. Ask for it at your favorite computer or book store.

By now, you're starting to see a lot of familiar things in the module. ResEdit means *resource editor* , all the resources you see can be edited with ResEdit. That's how you're going to create your own Kid's Clicks module.

ABOUT KID'S CLICKS & MODULES

When you create your own module, Kid's Clicks will be looking at your work to tell it what to do. Every module contains specific information for just that purpose. Kid's modules minimally contain the following resources, many of which you will edit to create your own:

- 1 'PICT' resource, ID# 5000
- 2 'DLOG' resources, ID#'s 1000 & 5000
- 2 'DITL' resources, ID#'s match 'DLOG' ID#'s
- 1 - ? 'snd ' resources, ID#'s 128 - ?
- 2 'STR ' resources, ID#'s -16396 & 5000
- 3 'vers' resources, ID#'s 1, 2, & 128

DESIGNING YOUR MODULE

Kid's Clicks allows users to click on a picture and hear corresponding sounds played. It's fun for kids, to hear sounds of animals, or silly sounds, and it can also be educational. Modules can teach, entertain, inform, or just about anything. A map with pictures of special things to click on related to each place

You Can Create Your Own Mac Program... continued

on the map; a family photo with satirical sounds to go with the faces; a chart representing important facts, with sounds to enliven the presentation. It's up to you, what do you want your module to do? Have an idea in mind? Well, let's get to work!

First, change the name of the copy you made of a Kid's Clicks module, to a name relative to your module.

You will need these items before you can begin assembling your module:

1. Find or create a picture, a graphic that we can use in a 'PICT' resource. You should find a picture that is fairly large, as close to 640 x 480 as possible. Kid's Clicks will scale your picture as needed for different screen resolutions.

2. You need some sounds. You can download sounds from online services and BBS's, or you can record them yourself. You'll have to explore the options to see what's best for you. There are many useful sound editing utilities available as freeware, shareware, and as commercial applications. Any double-clickable sound, or any file whose type is listed as "Sound" in your windows for folders in the Finder, will work.

These files contain resources called 'snd ' resources., which we can copy and paste into your module. Short sounds are better, because longer sounds take a lot of space to store in the module, and they take a lot of memory for Kid's Clicks to play them. Many sound applications will allow you to *down sample* sounds to shrink the number of bytes they use. The sound quality is reduced with down sampling, but frequently the difference isn't significantly noticable, and it may help to dramatically cut down the size of your module.

Once you've collected your sounds, you're ready to begin completing the module.

A Typical Sound File:



You Can Create Your Own Mac Program... continued

LET'S DO SOME DAMAGE!

Use your favorite graphics program to load your picture. Select the whole picture using "**Select All**" from its **Edit** menu (it should be there, if not you may need to determine another way to copy the picture). **Copy** the picture. **Open** your module in ResEdit, double-click the 'PICT' resource type, and then the 'PICT' resource whose ID# is 5000. **Paste** in your new picture. Did it work? Great. Now Save the module!

What can go wrong: You don't want to create a new 'PICT' resource. You want to replace the picture that is currently stored in 'PICT' ID# 5000 with your new picture. If you end up with a new 'PICT', click the new one and press the DELETE key, then go back over the directions and try again.

IS THAT IT?

Not quite, things get a little more complicated than this, but you shouldn't have any problem making it through the hoops. Just take your time and carefully follow directions.

EDITING DIALOGS

Close up the 'PICT' resource type and open the 'DITL' resource type. You should find 2 'DITL' resources; one we won't edit, just yet, numbered 1000, and one we will edit numbered 5000. These resources have corresponding resources called 'DLOG's. We won't need to worry about editing the 'DLOG' resources until you work on an "About" screen for your module.

Open 'DITL' #5000. ResEdit will bring up its 'DITL' editing window. Notice that the dialog now has your picture in it.

The 'DITL' editor lets you edit all of the items used in the dialog. In order to tell Kid's Clicks where a user has clicked in your dialog, we need to set up rectangular regions using an item called a "user item". This dialog already contains rectangular regions used by the previous module. Those regions will probably not be in sync with your module, so we will need to delete all of the user items used by the previous module.

You Can Create Your Own Mac Program... continued

Making it easy:

Choose “**Select All**” from the **Edit** menu (or hit Command-A). Every item in the ‘DITL’ will become an active selection. You can see that they have been actively selected by the moving outline around their rectangles.

Hold down the SHIFT key and click in the ‘DITL’ window in a place where there isn’t a grayed rectangle. (Those areas are the user items.) Now, the picture item’s rectangle should not be active, the moving outline around it should be gone; while all other items are still selected. If that isn’t the case, go back to “**Select All**”, and try again. If everything looks good, release the SHIFT key and hit the DELETE key. Presto! You have just removed every user item! You’re ready to begin creating your own user items.

Setting up your own user items:

You’ve probably noticed the floating palette next to the ‘DITL’ editing window. This palette contains different types of items that can be placed into a dialog. One of them is a user item.

Click on the palette where it says “User Item” and hold down the mouse button, then drag the cursor from the palette to your dialog. This will place a new user item into the dialog. You will add user items for each area you want to designate for users to click in to hear a different sound. Drag the user item you just created to a place on the picture where you want to have a sound played. In the bottom, right-hand corner of the user item is a small black box. This box will let you re-size the user item as needed, when you click in it and drag the desired direction.

Position and size your user item so that its rectangle encloses as much as possible the exact area you want to have a sound associated with. When it’s in position, double-click the user item. In the item window that comes up, check the box labeled, “Enabled”. Repeat this process for each area you have a sound to play. Do not add any item types to the ‘DITL’ other than user items.

Note: When people use your module, the lower item number will receive mouse clicks when items overlap.

You Can Create Your Own Mac Program... continued

A Final Adjustment

When you've added all the user items you need for each area in the dialog, select "**Select Item Number**" from the **DITL** menu, and select item #1. Choose "**Set Item Number**" from the same menu and change the item number to the highest number shown.

We had to change the pictures item number, otherwise it would prevent any of your user items from receiving mouse clicks. Save your module!

WHEW... YOU DID IT!

WIRED FOR SOUND!

Your ready for sounds now! This can be a tedious process, but once you get the hang of it, you'll be on your way in a flash.

On a piece of paper, make a list of the user items numbers in your dialog. Next to each number write down a name for the sound you want to correspond to the location of each user item number in the dialog. If the item numbers aren't showing, select "**Show Item Numbers**" from the **DITL** menu. When you're done making your list, close up the 'DITL' editor and any other associated windows that are open.

Your sounds may be in just one file, or in several, whichever it is, begin going down your list of sounds. Open the file containing the first sound on your list, with ResEdit. Remember: The sounds you are going to use must be in 'snd ' resources.

Open the 'snd ' resource type in the file containing the sound you want to use. Select the sound that you are going to use and **Copy** it. If you like, you can play the sound choosing options under the **snd** menu whenever a 'snd ' resource is highlighted. **Paste** the 'snd ' into your module. Repeat this process for each sound in the file you opened; you can also use "**Select All**" to **Copy** and **Paste** all of the sounds at once. Close the file you were copying sounds from, then proceed down your list until you have pasted all of the sounds you need into the module.

What can go wrong: If you receive a dialog asking if you'd like to replace resources with the same ID#, click the button that says "Unique ID".

When you've pasted into the module all of the sounds you want to use from all of the files that contained them, save your module!

You Can Create Your Own Mac Program... continued Mapping It Out

Now that you have all of your sounds, you can begin adjusting the ID#'s of the 'snd ' resources to correspond to the list you made of your user items. You will highlight each 'snd ' resource and use the **"Get Resource Info"** command from the **Resource** menu to change each one's ID# and to make sure each is checked "Purgeable".

Here is what the window for editing resource info looks like:

Info for snd 9005 from my module

Type: **snd** Size: **53802**

ID: **9005**

Name: **Kill duh wabbit!**

Owner ID:

Sub ID:

Owner type

DRVR **WDEF** **MDEF**

Attributes:

☐ System Heap ☐ Locked ☐ Preload

☒ Purgeable ☐ Protected ☐ Compressed

Change the ID# of each 'snd ' using this method:

ID# = 9000 + the user item number it corresponds to.

What can go wrong: You may run into conflicts if a 'snd ' resource already exists with the ID# you need. In that case, first find and change the other 'snd ' resource's ID to a number that you will not be using, or to the number it will have when you're done, before you change the current 'snd ' resource to the number.

Check the box labeled, "Purgeable" for every 'snd' resource!

When you are done— save your module!

You Can Create Your Own Mac Program... continued

ALMOST HOME!

If you did everything correctly, at this point you could launch your module and see your picture and have your sounds played when you click on the locations you set up. Close your module in ResEdit and try it out!

Did everything work? No?! Well, see the end of these directions for: trouble shooting tips.

If everything works, you're ready to complete your module. Open up your module with ResEdit to finish the job.

STRINGS

Your module contains a resource type called, 'STR '. This stands for "string". A string to a computer is a string of characters, like letters and numbers. Open up the 'STR ' resource type and double-click the resource whose ID# is 5000. This is the string Kid's Clicks will place in the Apple menu for users to select when they want to see a dialog that tells them about your module. Change the string so that it uses the name of your module. Save your module!

VERSION INFORMATION

Open the 'vers' resource type. You will need to edit resource ID#'s 1 and 2, and possibly 3. Open up the 'vers' resources numbered 1 and 2. These resources are used by the Finder to display information about your module through the "**Get Info**" command, and in your folder windows. I won't go into a lot of detail about doing this, but just try to adjust the strings in the 'vers' resources according to what you feel seems to work according to the format you see from the previous module; using appropriate information about your module.

When you've done that, open up 'vers' resource #3. If your module is designed to work with a version of Kid's Clicks different than the version numbers shown, change the numbers to those that are appropriate. Save your module!

You Can Create Your Own Mac Program... continued

YOUR “ABOUT” SCREEN

One last item to complete, your “About” screen. Open up the ‘DITL’ resource type and double-click on ‘DITL’ ID#1000. This will bring up the ‘DITL’ resource from the former module’s “About” screen. Delete every item number, except item #1. Item #1 must be a button, it should say, “OK”.

Now, you can design your own “About” screen, however you like it; except it cannot contain any other controls, like buttons, checkboxes, radio buttons, custom controls, or editable text. You can, however, add static text items. Opening up those items will let you type in whatever text you want it to say. You’ll need to re-size the item’s box to fit your text. You can also add icons and pictures. The rectangles of picture items will have to be set to fit your pictures. You can get exact dimensions by pasting them into the Scrapbook (at least in System 7.5’s Scrapbook), or you can use a drawing program with the capability to tell you the dimensions.

To set up a picture on your “About” screen, first add another picture to your module in a ‘PICT’ resource. Open up the ‘PICT’ resource type and select “**Create New Resource**” from the **Resource** menu; or type Command-K, to create a new ‘PICT’ resource. Paste your picture into the new ‘PICT’ resource you created. Make sure to check it as Purgeable! When you’re done with that, add a Picture item into the ‘DITL’ for your “About” screen. Double-click on the item to open it up. Set the resource ID# to the ID# of the ‘PICT’ resource you added. To set the dimensions for the picture, I find it easiest to set the Top and Left coordinates to 0 (zero) and then enter the picture’s width as the Bottom coordinate, and the height as the Right coordinate, then, I just drag the picture to the right place in the ‘DITL’ window.

To use icons on your screen, create a ‘CICN’ (color) icon resource and an ‘ICON’ (B&W) resource with the same ID#. Add an Icon item to your ‘DITL’. Open it and set the resource ID# to the number of your matching ‘CICN’ and ‘ICON’ resources.

When you’ve finished your “About” screen ‘DITL’, save your module!

THE ‘DLOG’

At this point, you will probably need to resize the dialog window to fit your “About” screen. Close up the ‘DITL’ windows you have open, and open up the ‘DLOG’ resource type. Select ‘DLOG’ #1000. ResEdit will present you with a screen that contains a window which resembles your desktop, called a miniscreen. Inside that window is a miniaturized picture of your “About” screen window.

You Can Create Your Own Mac Program... continued

Pull down the **Miniscreen** menu and set the miniscreen's size to "**512 x 342 - Classic**". Your "About" screen window will have to fit in the smallest resolution used by Mac monitors. This setting will change the miniscreen to show how much space your dialog window will use on monitors with the smallest screen resolution. If it appears to be too big to fit, you will need to do some shuffling in your 'DITL' to bring it down to size.

Since the 'DLOG' and the 'DITL' for your "About" screen have the same ID#, we can have ResEdit set the size of your dialog window to the size of your 'DITL' window, for you. Double-click on the miniature window depicting your dialog. This will bring up the 'DITL' editor again. Now, in the lower right-hand corner of your 'DITL' window is a small black square, like the user items had, to resize the window. Size the window so that it fits your 'DITL'. If your 'DITL' window is too big, edit it to bring it down to size. When your done, close the 'DITL' window and go back to the 'DLOG' window.

In the 'DLOG' window, check to see that your dialog window fits on the miniscreen. You can preview how your "About" screen will look like by selecting "**Preview at Full Size**" from the **DLOG** menu. If you need to, continue to make adjustments to your 'DITL' until everything seems right. Save your module!

FINAL CHECK

It is important that every resource in your module have its "Purgeable" box checked. Any resources you have added should be examined to be certain this is true. Kid's Clicks will run out of memory if you fail to check all the resources as purgeable. When you're done, save the module, AND make a back up copy!

TROUBLE SHOOTING

Yes, Murphy's Law is bound to strike you somewhere. If you're having problems, check these things to see if they might be causing the problem:

Are all of your user items Enabled?

Are all of your resources marked Purgeable?

Are all of the resource ID#'s set correctly according to directions?

Are any of your sounds an exceptionally large size in bytes?

If so, Kid's Clicks may not have enough memory to play them.

Are you missing any resources from the minimum required list?

Did you change the module file's type?

The file information for your module should have the type listed as "Kmod" and the creator as "Kydz".

You Can Create Your Own Mac Program... continued

Go over the directions and examine the trouble shooting tips carefully. Hopefully, you'll be able to work out any problems successfully and your module will be complete.

YOU DID IT!!

Congratulations! You are now a Mac programmer! You've created a program unique to your design. Now, call your friend Joe PC and show him how computer literate you are!

There's much that can be done with ResEdit, alone; without hundreds of dollars of software and reference materials. You can customize programs to your hearts content. Be wary, however, you never really know if a change will cause a negative side effect, always work on a copy! Save your original unmodified files. Also, remember that while you can modify anything you own, you cannot violate someone else's copyright by distributing a modified version of their file. Kid's Clicks modules that you have created can be distributed, so long as items you have added do not violate copyrights themselves. Modified versions of Kid's Clicks itself, may not be distributed. All of us benefit from this kind of quality control.

THE END.

"You Can Create Your Own Macintosh Program— With ResEdit!"

By Eric Long, ©1995

This article may be reprinted with permission only. Permission is hereby granted, up receipt of 1 copy of publication containing reprint.

Correspondence: Email: EricMLong@aol.com
CompuServe - 72764,1072
Post: 4020 Jeffrey Avenue
Sacramento, CA 95820-2551

"Kid's Clicks!" ©1995 Eric Long
ResEdit™ © Apple Computer Inc.

Legal Disclaimer:

Neither ResEdit, nor Kid's Clicks, or its modules, can ever be guaranteed to work without fault. Problems can and do occur, use solely at your own risk.