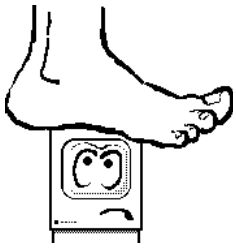

AMUG Tenderfoots #7

Stepping Into Scripts

By L. Frank Turovich

Last month we began creating a simple database stack to maintain a collection of audio CDs, tapes, and records. In it we talked about creating buttons and fields in the background layer of our stack. For those that missed last month Figure 1 shows the final layout for the Records stack. Note that all fields, buttons, and graphics exist on the background layer of the stack.

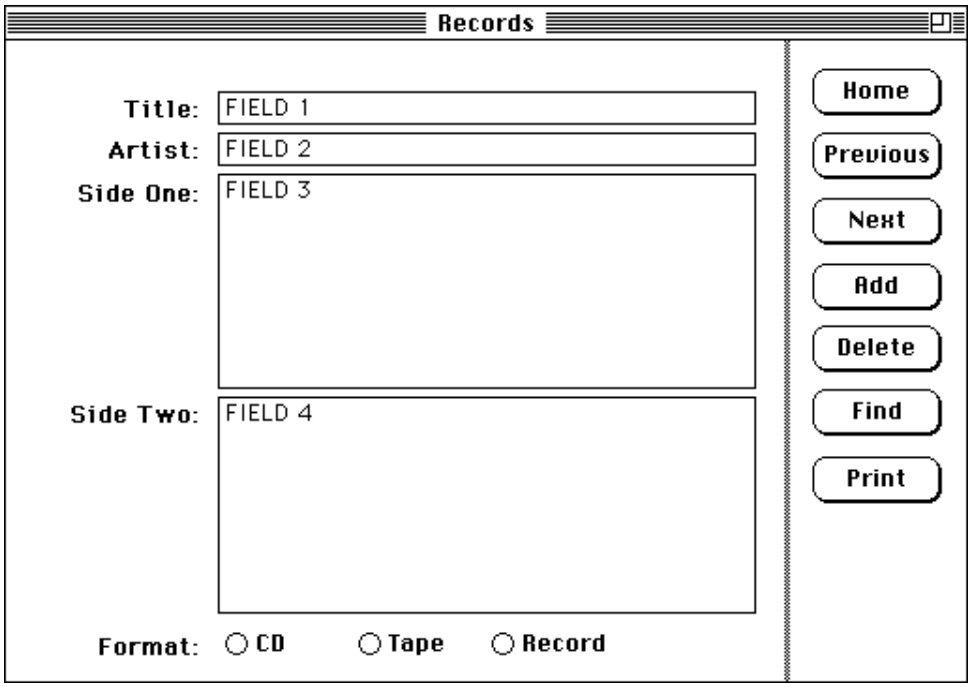

Figure 1. Layout for Records stack.

This month we add some functionality to the stack by adding scripts to the various buttons. To create scripts, we need to enter the HyperCard script editor. There are two methods of doing this: the slow way, double-click on the button itself, then selecting the Script button in the Button Info dialog. The fast way, hold the Shift key down while double-clicking on a button. You are immediately taken to the script editor. A trick when you are done using the script editor, hit the Enter key. This automatically saves any changes avoiding the very tedious appearance of the save script dialog.

What is Scripting?

Well, onto the scripting. What is scripting you may be asking yourself. Scripting is HyperCard's method of programming the Macintosh. It involves the writing of commands, a collection of statements that describe to the computer how to execute a series of actions. When creating scripts we will start simple and build from there.

Scripts are like a cooking recipe, a little of this, a bit of that, all in correct proportions and you a have a feast fit for a king. A little too many ingredients or skip a few ingredients and the feast becomes a fast. A script is something like that. Tell the computer exactly how to perform a a series of actions, all in proper order, and you have a working program. Leave something out, or give it imprecise directions, and it can refuse to operate, or worse yet, do something totally unexpected.

The references I use include the reference manual that came with HyperCard and Danny Goodman's fine book on HyperCard. There are many others just as useful. If you have one don't hesitate to use it frequently while you are learning. Unlike other Macintosh programs, using a programming language sometimes requires quite frequent referals to the manual for syntax and usage information.

Button Scripting

We'll start with the top button and work our way down. If you followed the directions given last month the Home button already works to move you back to the Home stack. If you didn't, there's another way to add functionality to the Home button.

First, open the Home button for scripting. Hold the Shift key down and double-clicking on the Home button. You should see something like Figure 2 if you followed last months directions. If not, the button script will be empty.

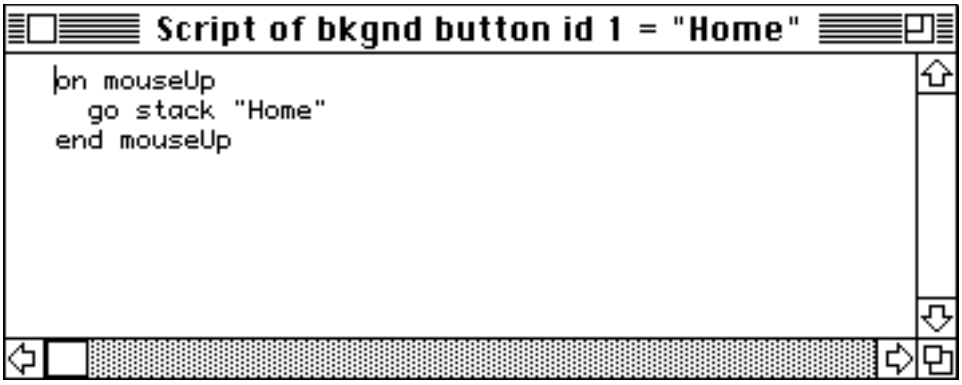

Figure 2. Home button script.

The first you thing you should notice is the ON MOUSEUP statement. This is called a message handler. MOUSEUP is a message produced by the mouse whenever the mouse button is released. If the mouse up occurs while the mouse is positioned over the button Home, the MOUSEUP message is sent to that button. If the mouse button is released over a button (or any other object for that matter) that doesn't have a MOUSEUP handler, the message is ignored.

The handler traps the MOUSEUP message. It tells HyperCard in effect, I will handle this message myself. If a message isn't trapped by an object, the message is passed up a hierarchy of objects all the way to HyperCard itself. At any point along the way the MOUSEUP message can be intercepted and handled by another ON MOUSEUP handler. Other messages are also possible including: MOUSEDOWN, MOUSESTILLDOWN, MOUSEENTERING, MOUSELEAVING, and many others. Refer to your HyperCard manual or book for a complete list of messages.

The next thing you see is the HyperTalk commands "go". This tells HyperCard to transfer control to another card or stack. In this case we need to go to a stack so the modifier "stack" was appended. We also need to identify the stack to go to so we use the stack name, in this case "Home" as the identifier. So the completed script is:

Home Button:

```
ON MOUSEUP
    GO STACK "HOME"
END MOUSEUP
```

which means, transfer control to the stack called Home. Note the use of quotes around the stack name. When identifying a card or stack by name you should always enclosed the name in quotes. That way, names like "Fred's Stack" will be identified by HyperCard as a name and not a variable.

So, if you haven't entered the script for the Home button yet, enter if now. When done, press the Enter key to automatically save your work. Now try the Home button. It should take you back to the Home stack without any problem. Congratulations, you've entered your first successful script.

Three other buttons also make use of the GO command, the Previous, Next and Add buttons. In two cases, however, the GO command moves you forward or backward in the cards we expect to have later in the stack database, the third takes us to the last card in the stack. Open each button in succession and enter their respective scripts:

Previous Button:

```
ON MOUSEUP
    GO PREV CD
END MOUSEUP
```

Next Button:

```
ON MOUSEUP
    GO NEXT CD
END MOUSEUP
```

Note the use of the abbreviation PREV for PREVIOUS and CD for CARD. There are dozens of useful abbreviations that make writing scripts very fast and not so typing intensive. Also, note that next and previous enable us to go to any card in the stack, as long as it is either before or after the current card. You can also create scripts that direct you to a specific card by identifying it more precisely. We will see how this is done in the next script.

Now we need to create some cards to test our Previous and Next buttons. Open the Add button and enter this script:

Add Button:

```
ON MOUSEUP
        GO LAST CD
        DOMENU "NEW CARD"
END MOUSEUP
```

This script will first position HyperCard at the last card in the stack with the GO command. We go to the last card since we don't want to add new cards in the middle of our stack, only at the end. It then executes the DOMENU message to implement the "NEW CARD" menu item. You can find this item under the Edit menu in HyperCard. DOMENU enables you to access and use all of HyperCard's menus quickly and easily just by using their titles (always use quotes for strings). Close the Add button scripting window after saving your work.

Now test all of your buttons. To know that you change cards (since there are no reference points that change) scrawl a little number with the graphic pencil onto the current card. Now select the Add button. The number should disappear. Scrawl another number and hit the Add button a second time. The last number should disappear also. Scrawl a third number. Now test the Next and Previous buttons. Each number should appear as you maneuver through the three cards in the Records stack.

That's all my room for this month, next month we will tackle some more intense scripting. Until then, good scripting. Frank