



UNIVERSITY OF UTAH

M a c i n t o s h M a n a g e r M e e t i n g



iHook

By

James Reynolds

University of Utah

Student Computing Labs

What is iHook?

- iHook is a graphical front end for any commandline executable
- Your scripts or UNIX tools can provide user feedback in an aqua window
- Much easier than writing a Cocoa NSTask wrapper or an AppleScript Studio application

Who develops iHook?

- Research Systems Unix Group



- University of Michigan
- The same guys that make radmind
- The main developer is Andrew Mortensen

Contact & links

- Bug reports, feature requests, other feedback:
 - ihook@umich.edu
- Webpage:
 - <http://rsug.itd.umich.edu/software/ihook>
- iHook is open source
 - <http://rsug.itd.umich.edu/cgi-bin/cvsweb.cgi/>
- iHook script archive
 - <http://rsug.itd.umich.edu/software/ihook/hooks.html>

History of iHook

- Started out as "RMLogout".
 - It simply executed the radmind tools
 - ktcheck
 - fsdiff
 - lapply
- Radmind administrators complained about the lack of customization in RMLogout, especially in error handling
- RSUG changed RMLogout to iHook

Why is iHook important?

- It allows admin to “lock out” users while maintenance is running
- iHook is the easiest way for a script to give user feedback (opposed to Cocoa or AS-S)
- Because any script can be run, it is a most excellent tool
- It is currently the only way to have a screen saver over the login panel

iHook vs. script

- iHook is a Cocoa app that displays a window
- iHook executes a script or unix tool
 - “Script” refers to the script/tool iHook executes
 - iHook does not provide the “script”
 - You write your own script
 - Simple sample scripts provided with iHook

How to launch iHook

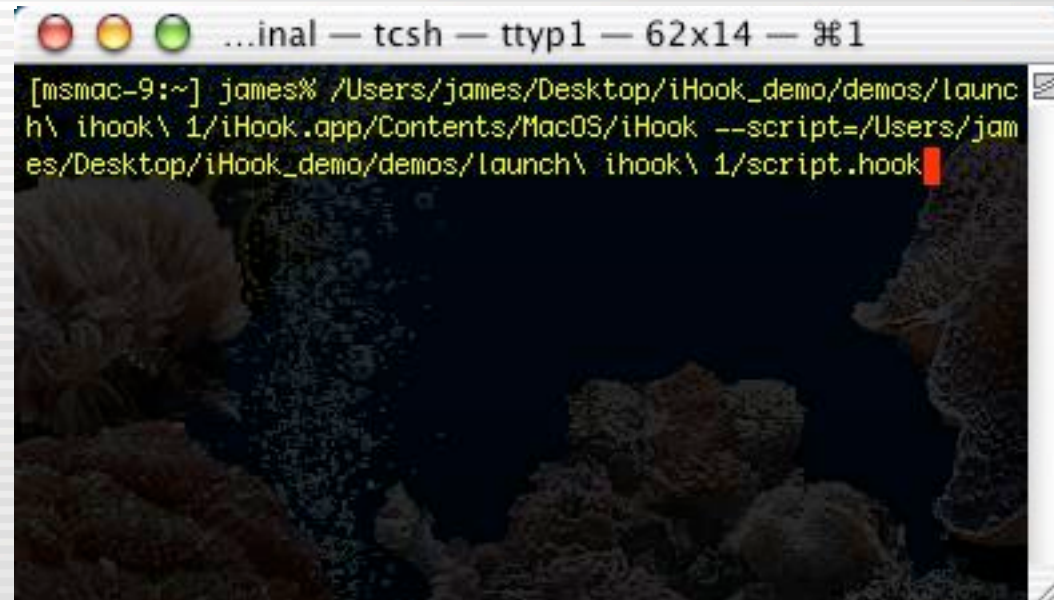
- iHook can be launched several ways
 - Script/Command line
 - Double click the application
 - Double click a file that ends with .hook
 - Drop an executable (script or tool) on iHook
 - LogoutHook
- Depending on how you launch iHook, iHook behaves differently

How to launch iHook

- Script/Command line
 - `/path/to/iHook.app/Contents/MacOS/iHook`
`--script=/path/to/script`
- LogoutHook
 - iHook tries to execute `/etc/logout.hook`
- Or the Logout/LoginHook (or even a StartupItem) could execute a script that launches iHook w/ the “`--script=/path/to/script`” parameter

How to launch iHook

- Demo

A screenshot of a Mac OS X terminal window. The title bar reads "...inal — tcsh — ttyt1 — 62x14 — ⌘1". The terminal text shows a user named 'james' at a prompt '[msmac-9:~]' entering the command: `/Users/james/Desktop/iHook_demo/demos/launch\ ihook\ 1/iHook.app/Contents/MacOS/iHook --script=/Users/james/Desktop/iHook_demo/demos/launch\ ihook\ 1/script.hook`. The background of the terminal is a dark, textured image of coral or rocks.

```
[msmac-9:~] james% /Users/james/Desktop/iHook_demo/demos/launch\ ihook\ 1/iHook.app/Contents/MacOS/iHook --script=/Users/james/Desktop/iHook_demo/demos/launch\ ihook\ 1/script.hook
```

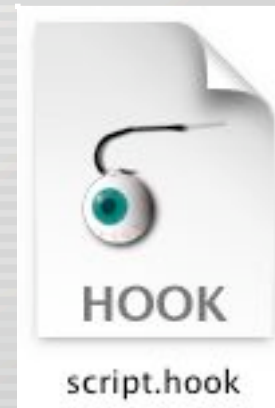
```
console "/System/Library/CoreServices/loginwindow.app/Contents/MacOS/loginwindow  
-LogoutHook /path/to/iHook.app/Contents/MacOS/iHook" vt100 on secure  
window=/System/Library/CoreServices/WindowServer onoption="/usr/libexec/getty  
std.9600"
```

How to launch iHook

- Double click the application
 - iHook prompts the user to choose a script/tool to run
- Double click a file that ends with .hook
 - iHook tries to execute the file
- Drop an executable (script or tool) on iHook
 - iHook tries to execute the file

How to launch iHook

- Demo



How to launch iHook

- Logged in?
 - iHook's window level is set to `NSNormalWindowLevel`
- Logged out?
 - Window level is `NSScreenSaverWindowLevel - 1`

How to launch iHook

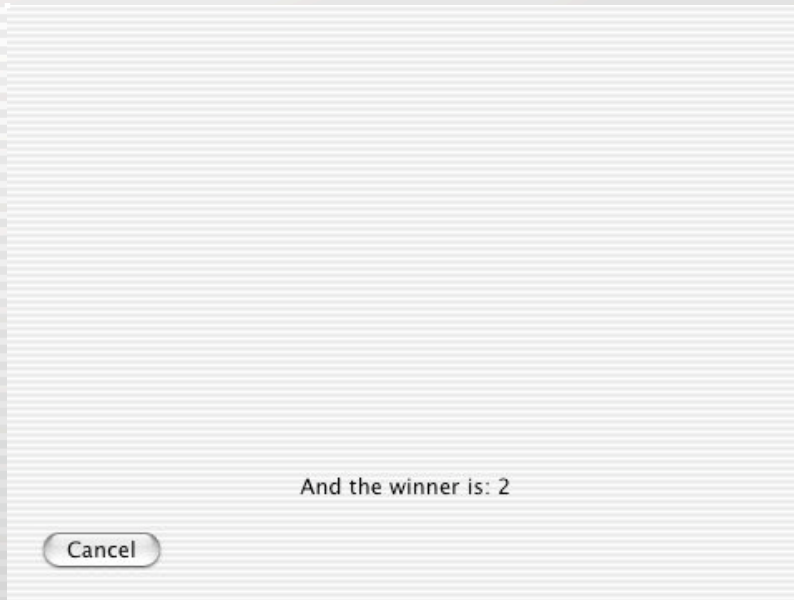
- Sudo?
 - iHook must be super-user to give a script super-user permissions
- Does iHook send the script any arguments?
 - No, but...
 - There is a \$CONSOLE_USER environment variable

Script feedback

- iHook puts script stderr in a hidden drawer
 - Revealed with command + L
- iHook puts script stdout in the main window
 - Exception, lines beginning with "%"
 - These are understood to be iHook Directives
 - iHook directives manipulate the iHook window

Script feedback

- Demo

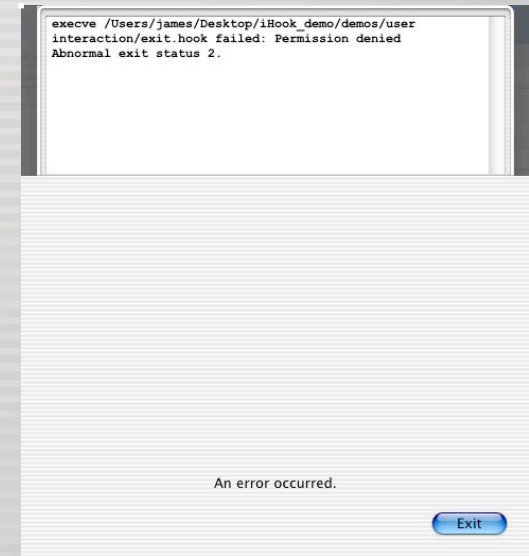


User Interaction

- iHook accepts only 3 events as user interaction
 - Command - L to toggle log window
 - Exit button when there is an error
 - Cancel button sends SIGTERM to script
 - Adding a SIGTERM trap to your script is the only way to get feedback from iHook into your script

User Interaction

- Demo



iHook Directives

- See “iHook Lexicon.txt” included with download
- New directives added in each version of iHook
- Directives do things, such as:
 - Control size, location, background graphic, and level of window
 - Set state of barberpole or progress bar
 - Show/hide a timer
 - Show/hide/disable Cancel button
 - More

iHook Directives

- Directives as of iHook 0.8.7:
 - %BEGINPOLE
 - %ENDPOLE
 - %<0-100>
 - %SHOWTIMER
 - %HIDETIMER
 - %OPENDRAWER
 - %CLOSEDRAWER
 - %BECOMEKEY
 - %RESIGNKEY

iHook Directives

- Directives as of iHook 0.8.7:
 - %BACKGROUND /path/to/an/Image
 - %WINDOWLEVEL <HIGH/NORMAL>
 - %WINDOWSIZE <WIDTH> <HEIGHT>
 - %WINDOWPOSITION <CENTER/TOP/BOTTOM>
<LEFT/RIGHT>
 - %CANCEL <ENABLE/DISABLE/REMOVE>
 - %TITLE Title String
 - %DEBUG

iHook Directives

- Demo



iHook window behavior

- The minimum window size is 200 x 200
 - Edit the NIB to make smaller...
- Exit button is removed at launch
 - Replaced and displayed if script exit value is > 0
- Changing window size centers the window

iHook exit value

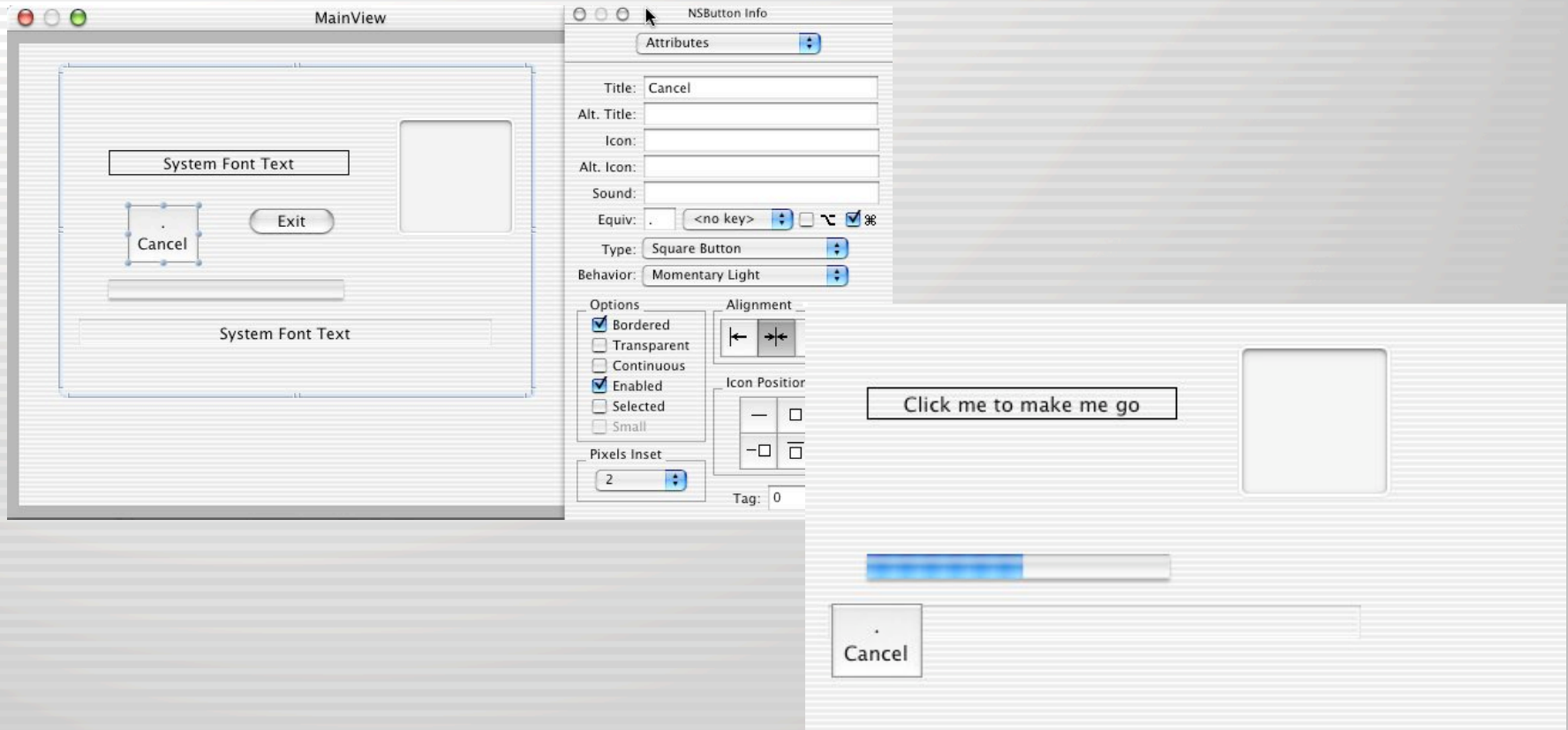
- iHook exits with the value of the script/tool
 - Useful to cancel login when run as the LoginHook
 - In 10.1.x, login canceled if LoginHook exited with non-0 value
 - This is broke in 10.2.x
 - Loginwindow never comes back

Customizing iHook

- Edit the NIB using Interface Builder
 - Change image location and centering rules
 - Change location, size of
 - Barberpole/progress bar
 - Timer
 - Other buttons
 - Window (to make smaller than 200 x 200)

Customizing iHook

- Demo



Customizing iHook

- Get source from:
<http://rsug.itd.umich.edu/cgi-bin/cvsweb.cgi/>
- Uses **execve** to launch the executable
- Uses **select**, **fgets**, and **read** to get output from stdout and stderr

Customizing iHook

- It is possible to replace the background image with a QT movie...
 - Change the imageview to a movieview in NIB and LHController.h
 - Use NSMovie's initWithURL:byReference: and NSMovieView's setMovie: instead of setImage
 -

iHook scripts

- There are 3 scripts included with iHook:
 - Backup script
 - Radmin script
 - Screensaver script
- These scripts are provided in perl and bash
- iHook script archive
 - <http://rsug.itd.umich.edu/software/ihook/hooks.html>

Writing scripts for iHook

- Print or echo the directives for barberpole/ progress bar and background graphic ASAP
- Always specify the full path to unix commands
- Learn bash redirection
 - “>”, “2>&1”, “>>”, etc.,
 - <http://www.tldp.org/LDP/abs/html/io-redirection.html>
 - tee command could also be helpful (see man page)

Writing scripts for iHook

- In perl, turn on autoflush feature, to prevent buffering
 - See sample perl scripts
- Add SIGTERM trap for “screensaver” scripts
 - See sample screensaver scripts

Writing scripts for iHook

- Screensaver demo
- Radmind demo

