Secure Automation Techniques

Automation meets Security in OS X

Session 206

Sal Soghoian
Product Manager Automation Technologies

Chris Nebel
Senior Engineer Automation Technologies

Introduction

Introduction

- Security is a focus of OS X

Introduction

- Security is a focus of OS X
    - App Sandbox

Introduction

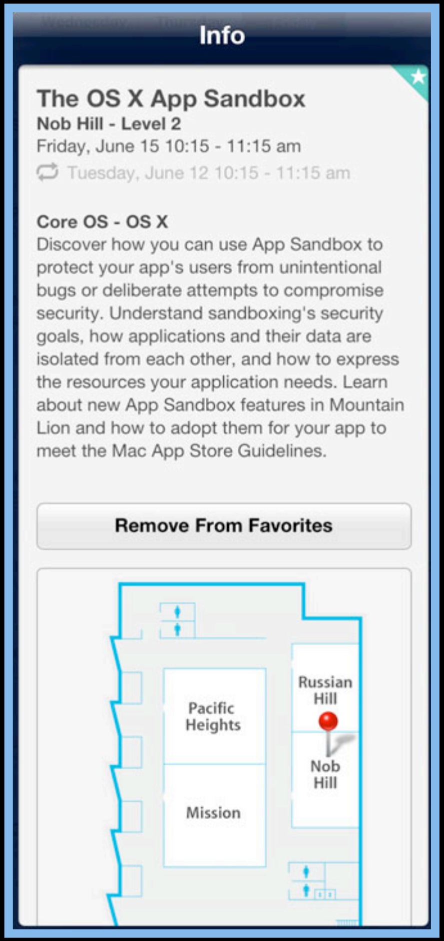
- Security is a focus of OS X
    - App Sandbox
    - Gatekeeper

Introduction

- Security is a focus of OS X
  - App Sandbox
  - Gatekeeper
- How automation works with new OS security designs

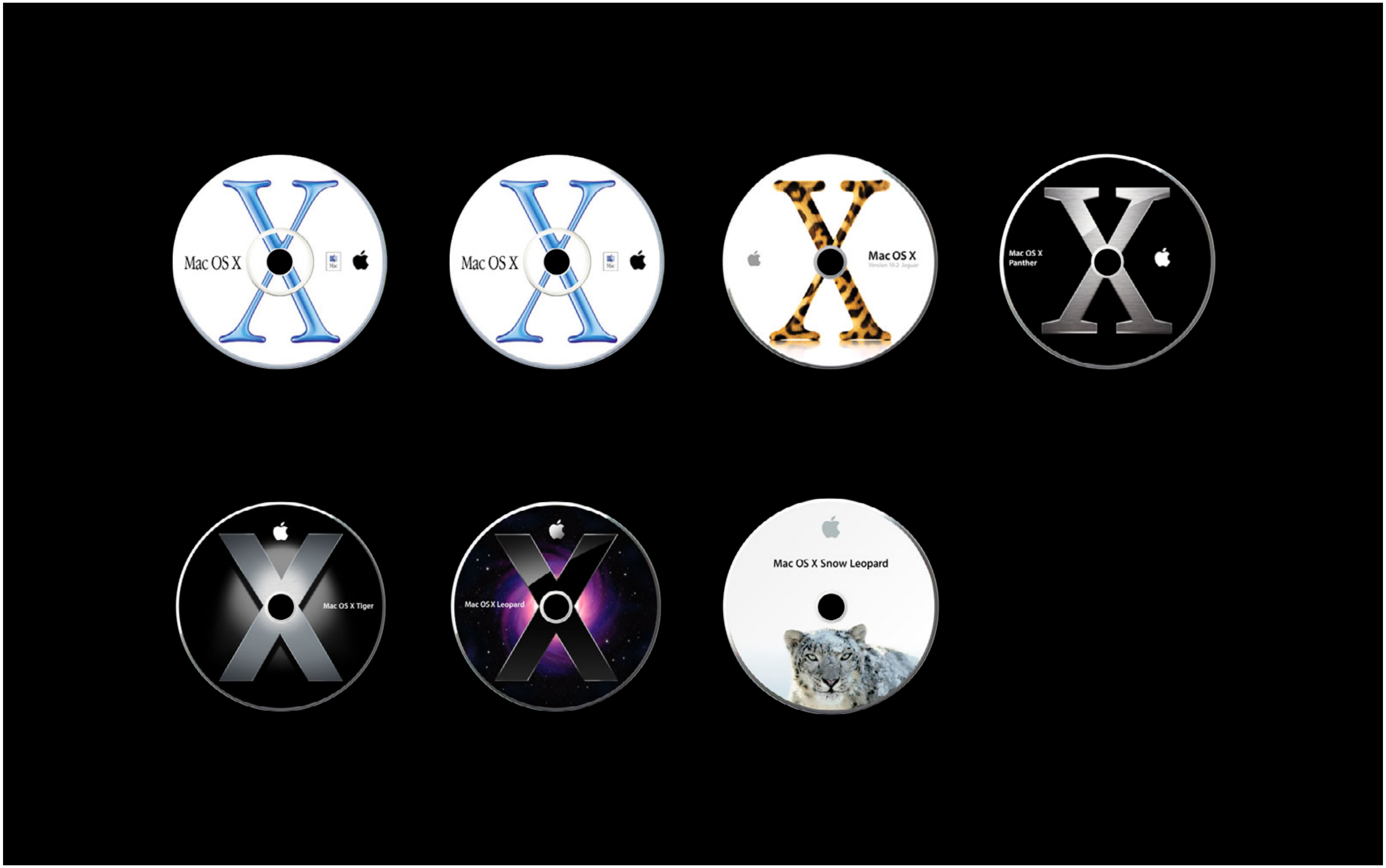
Introduction

- Security is a focus of OS X
  - App Sandbox
  - Gatekeeper
- How automation works with new OS security designs



The OS X App Sandbox
Nob Hill - Level 2
Friday, June 15 10:15 - 11:15 am
Tuesday, June 12 10:15 - 11:15 am

Core OS - OS X
Discover how you can use App Sandbox to protect your app's users from unintentional bugs or deliberate attempts to compromise security. Understand sandboxing's security goals, how applications and their data are isolated from each other, and how to express the resources your application needs. Learn about new App Sandbox features in Mountain Lion and how to adopt them for your app to meet the Mac App Store Guidelines.


Mac OS X
Mac
Mac OS X
Mac
Mac OS X
Version 10.2 Jaguar
Mac OS X
Panther
Mac OS X Tiger
Mac OS X Leopard
Mac OS X Snow Leopard

Desktop Services Installation

Cocoa-AppleScript Applets

Workflow Conversion

Video Encode Service

Script Templates

Web Content Actions

Text-to-EPUB Action

Desktop Actions Installation

Media Preparation Actions

Global Script Application Targets

AppleScript/Objective-C

Terminal Services

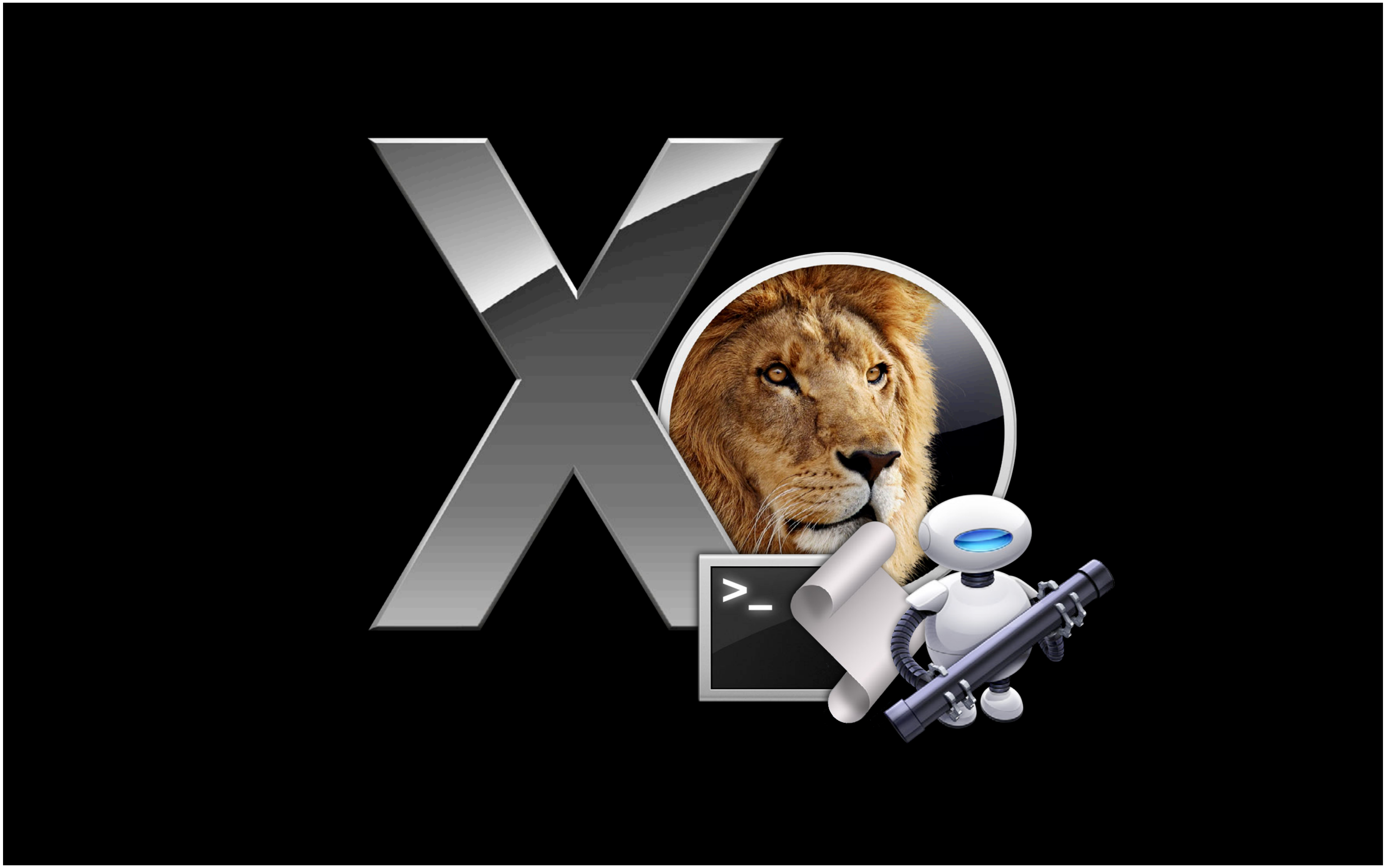
Workflow Versions

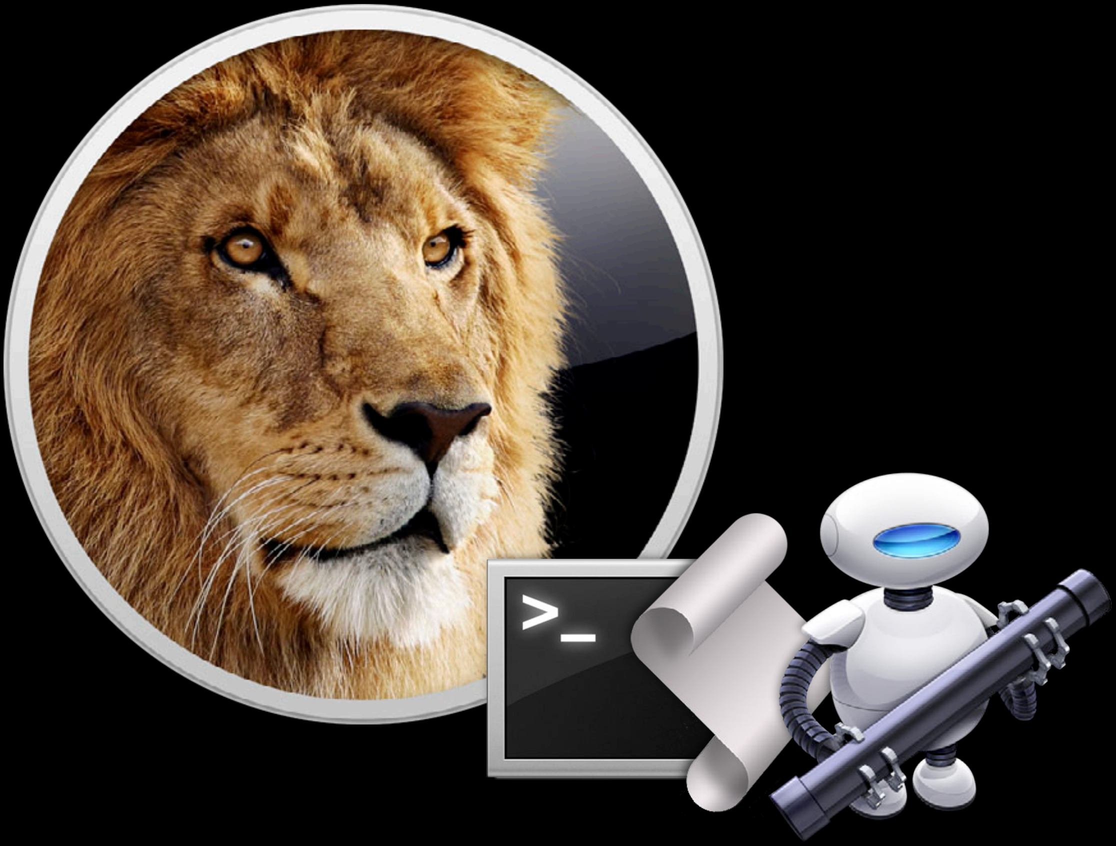
Website Popup Action

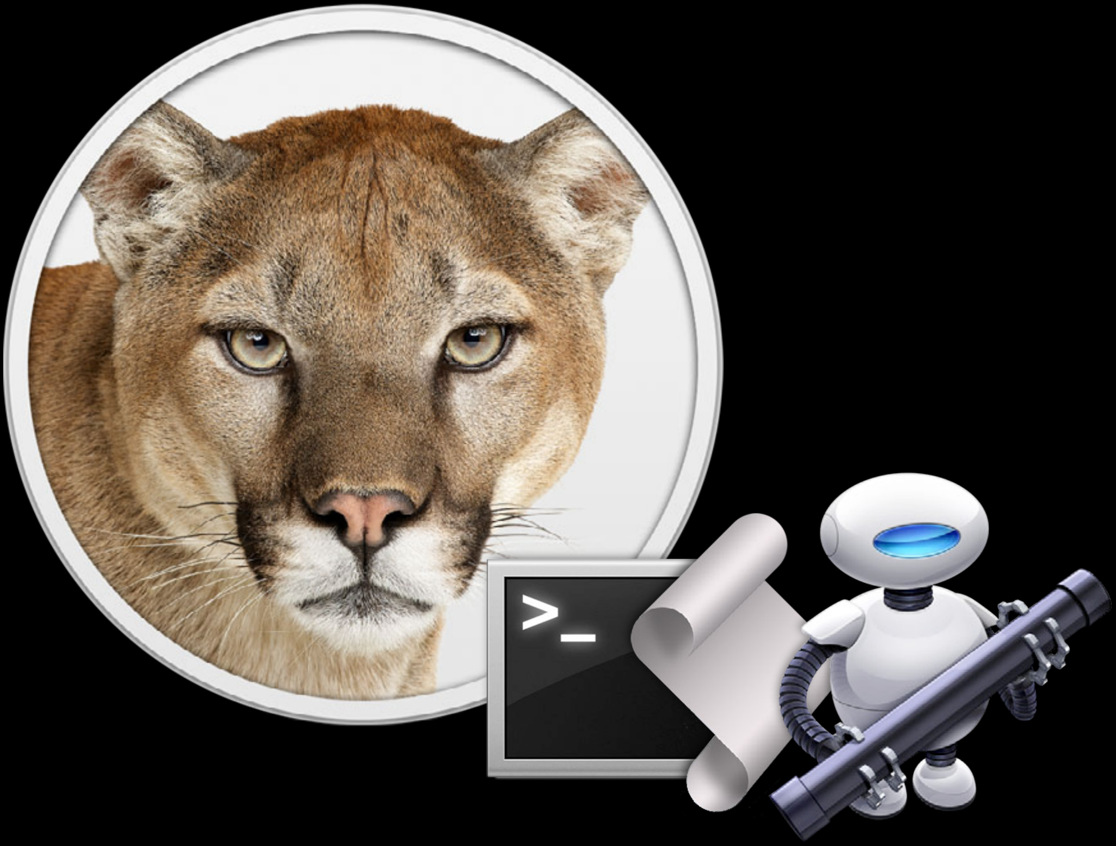
Audio Encode Service

Terminal Status Controls

Terminal Services for Finder



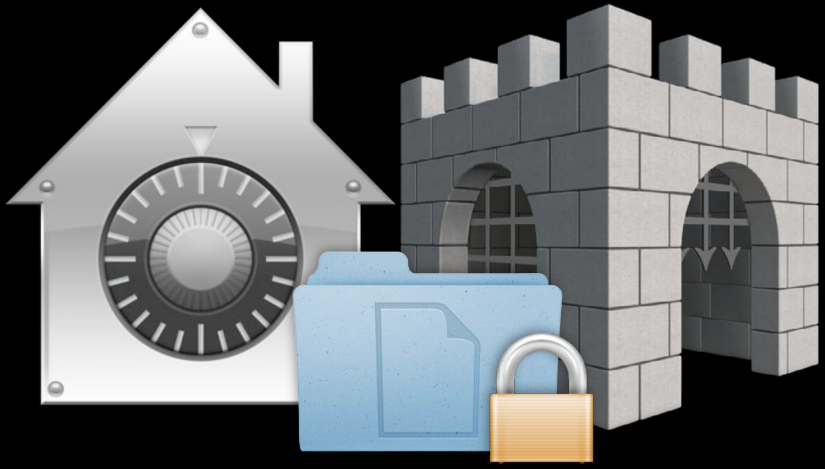




Mail 2
Calendar 4
Messages 1

OS X Mountain Lion

Security

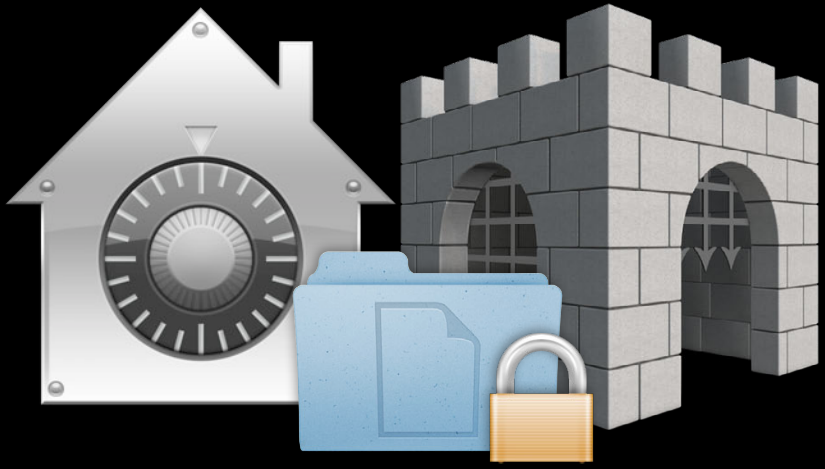


OS X Mountain Lion

Security

OS X Mountain Lion

Security

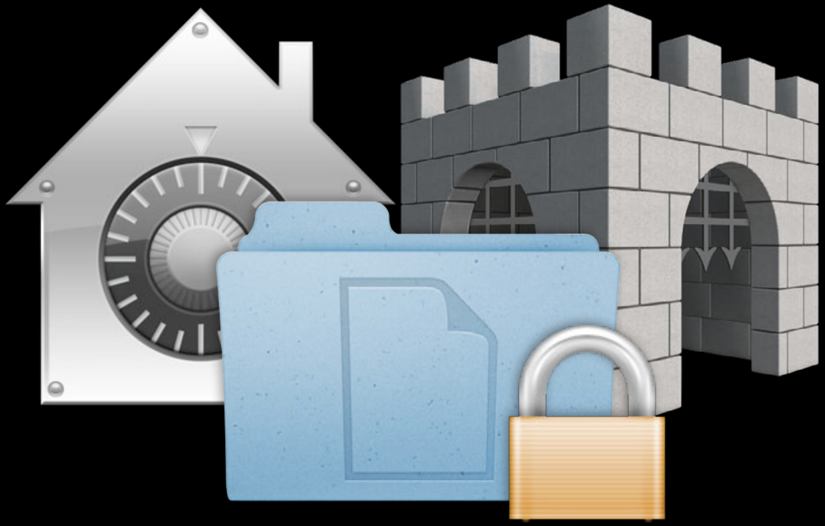


OS X Mountain Lion

Security

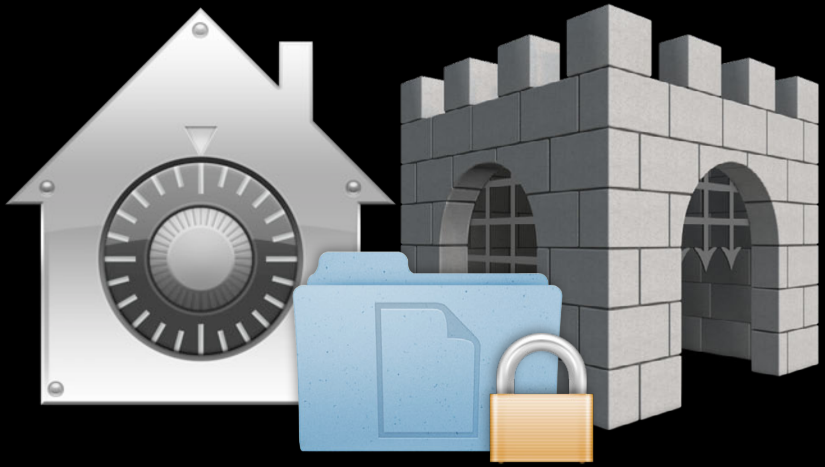
OS X Mountain Lion

Security



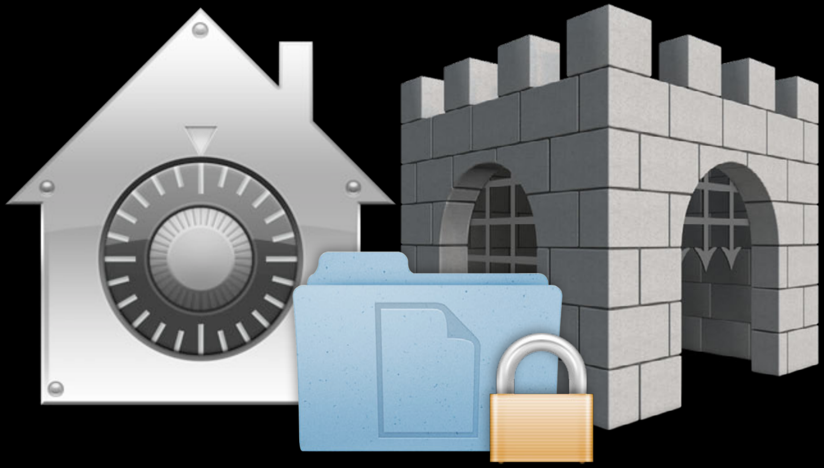
OS X Mountain Lion

Security

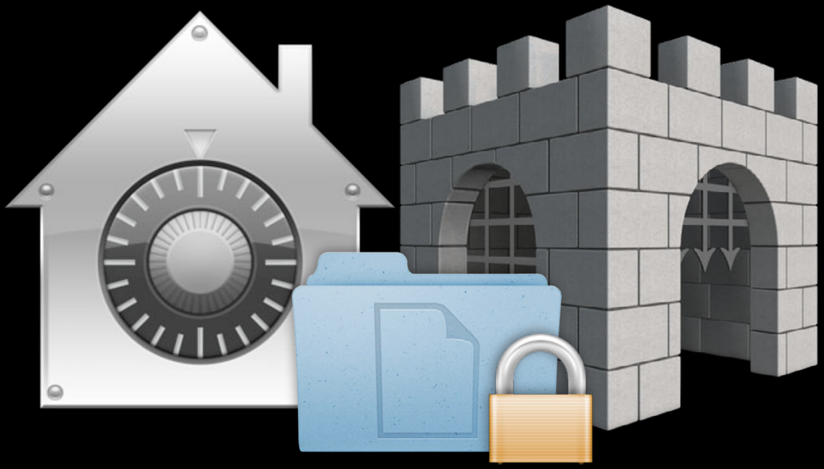


OS X Mountain Lion

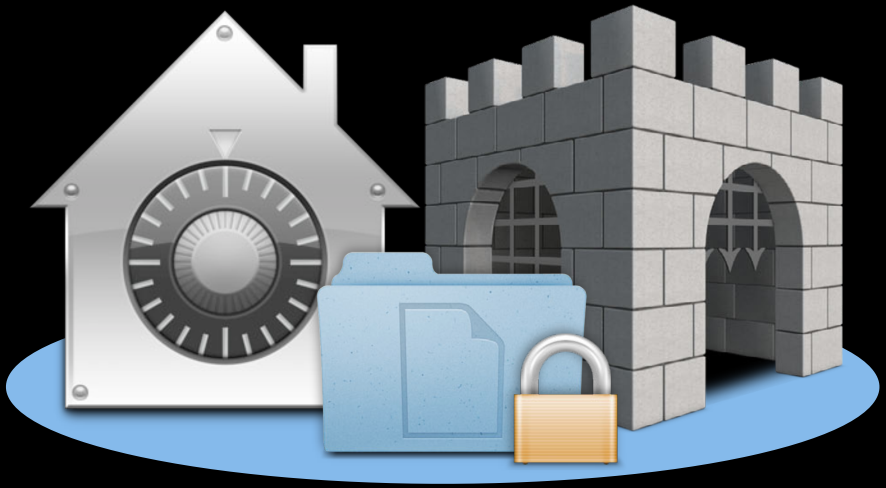
Security

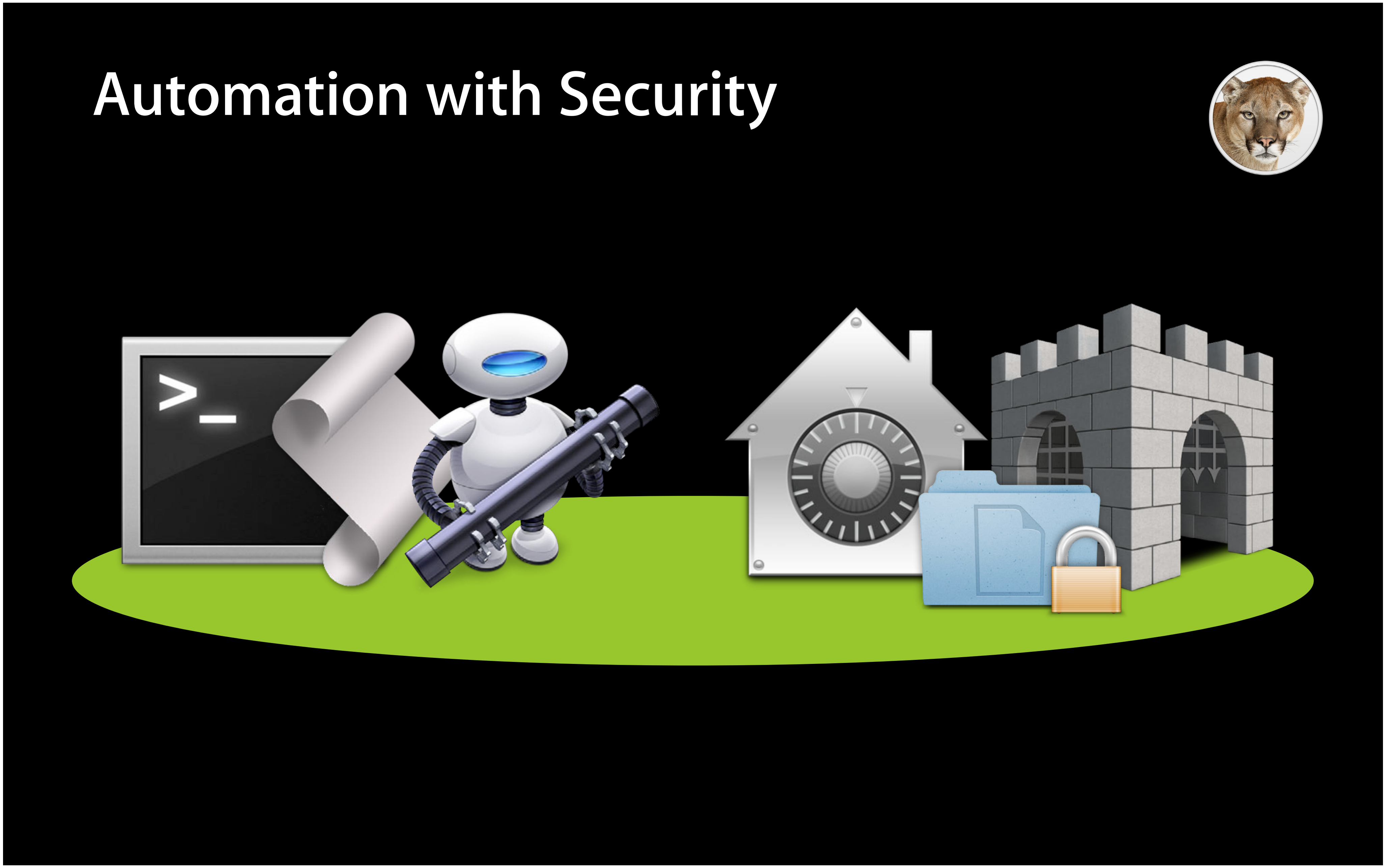


Security



Automation and Security





Automation with Security

Automation with Security

Automation with Security

Design goals

Automation with Security

Design goals

- Preserve functionality

Automation with Security

Design goals

- Preserve functionality
- Transparent interaction

Automation with Security

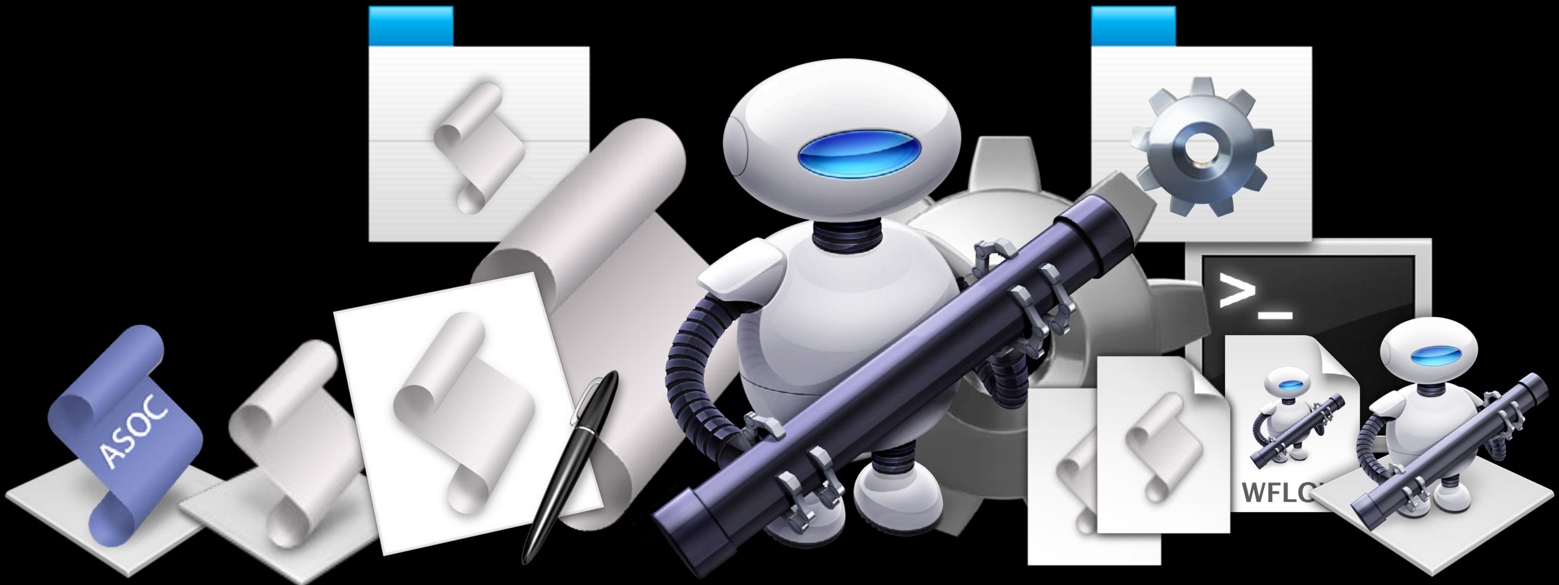
Design goals

- Preserve functionality
- Transparent interaction
- Minimize changes

[redacted]

4
Automation Scenarios

1. Personal Automation

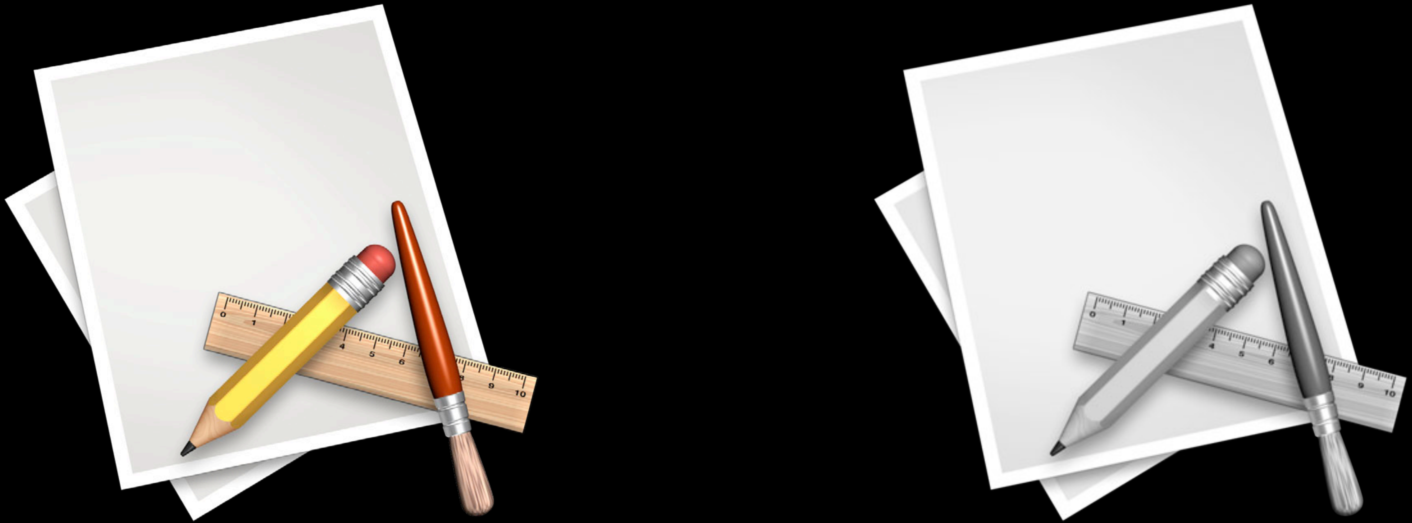


AppleScript Automator Services

2. Distributing Scripts

3. Application-to-Application Automation

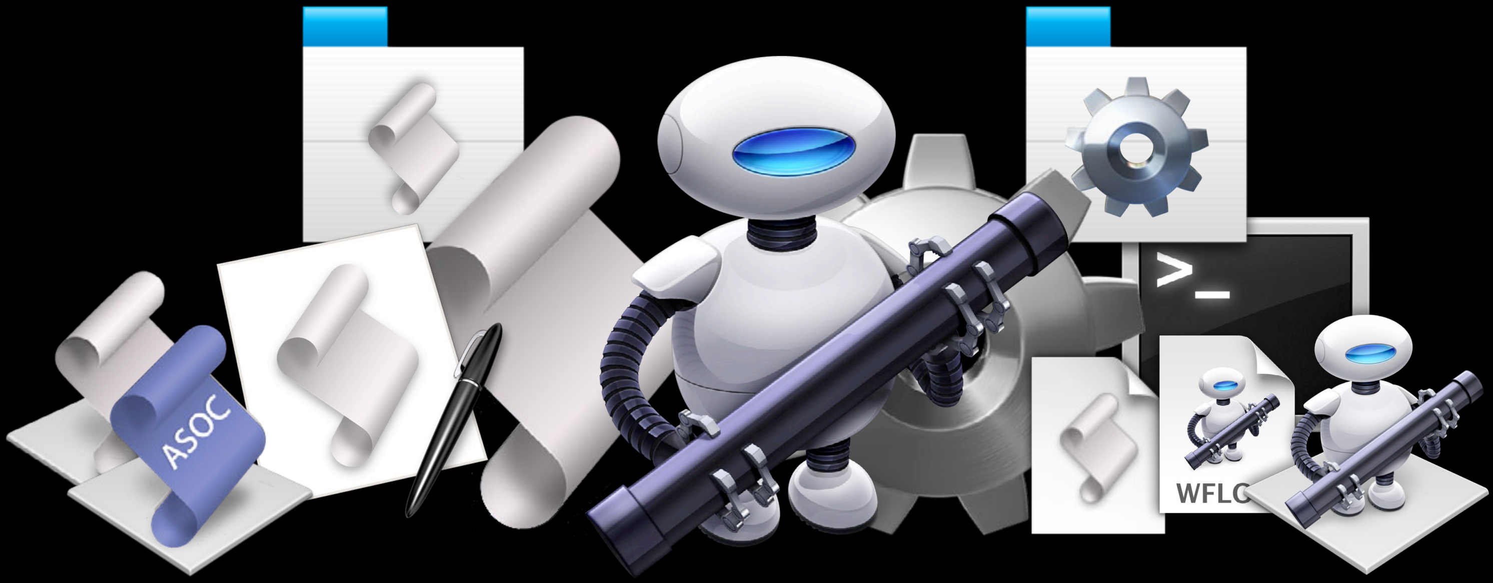
3. Application-to-Application Automation



4. Attaching Scripts

Personal Automation

Scripts you write for yourself



AppleScript Automator Services Terminal

Workflows, Scripts, and Applets

Automator

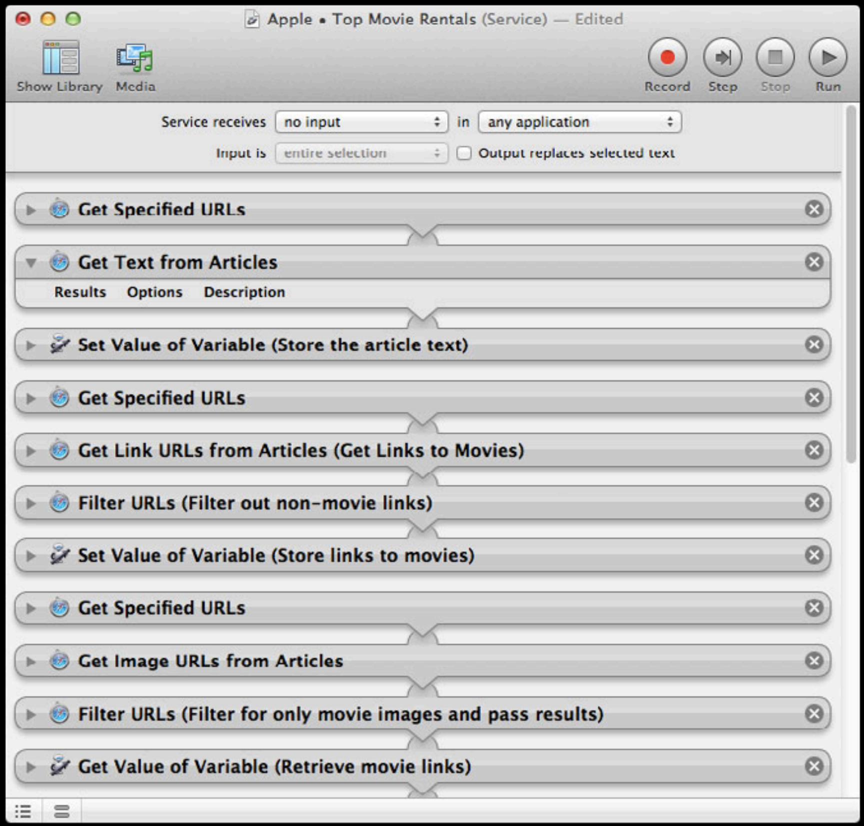
Workflows, Scripts, and Applets

Automator

"Otto"

Workflows, Scripts, and Applets

Automator


Apple • Top Movie Rentals (Service) — Edited

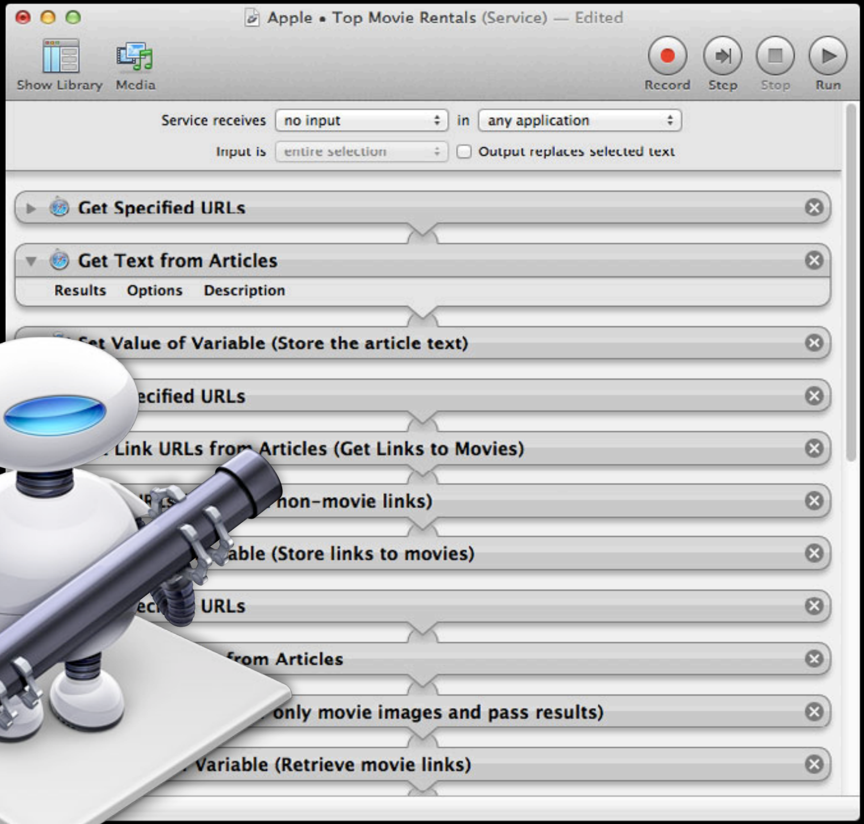
Service receives: no input in any application
Input is: entire selection
☐ Output replaces selected text

- Get Specified URLs
- Get Text from Articles (Results, Options, Description)
- Set Value of Variable (Store the article text)
- Get Specified URLs
- Get Link URLs from Articles (Get Links to Movies)
- Filter URLs (Filter out non-movie links)
- Set Value of Variable (Store links to movies)
- Get Specified URLs
- Get Image URLs from Articles
- Filter URLs (Filter for only movie images and pass results)
- Get Value of Variable (Retrieve movie links)

Workflows

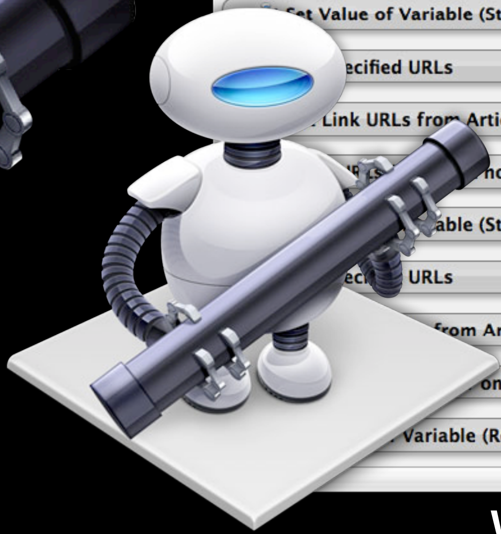
Workflows, Scripts, and Applets

Automator



Service receives: no input; in: any application
Input is: entire selection; Output replaces selected text [ ]

Get Specified URLs
Get Text from Articles
Set Value of Variable (Store the article text)
ecified URLs
Link URLs from Articles (Get Links to Movies)
non-movie links)
able (Store links to movies)
URLs
from Articles
only movie images and pass results)
Variable (Retrieve movie links)

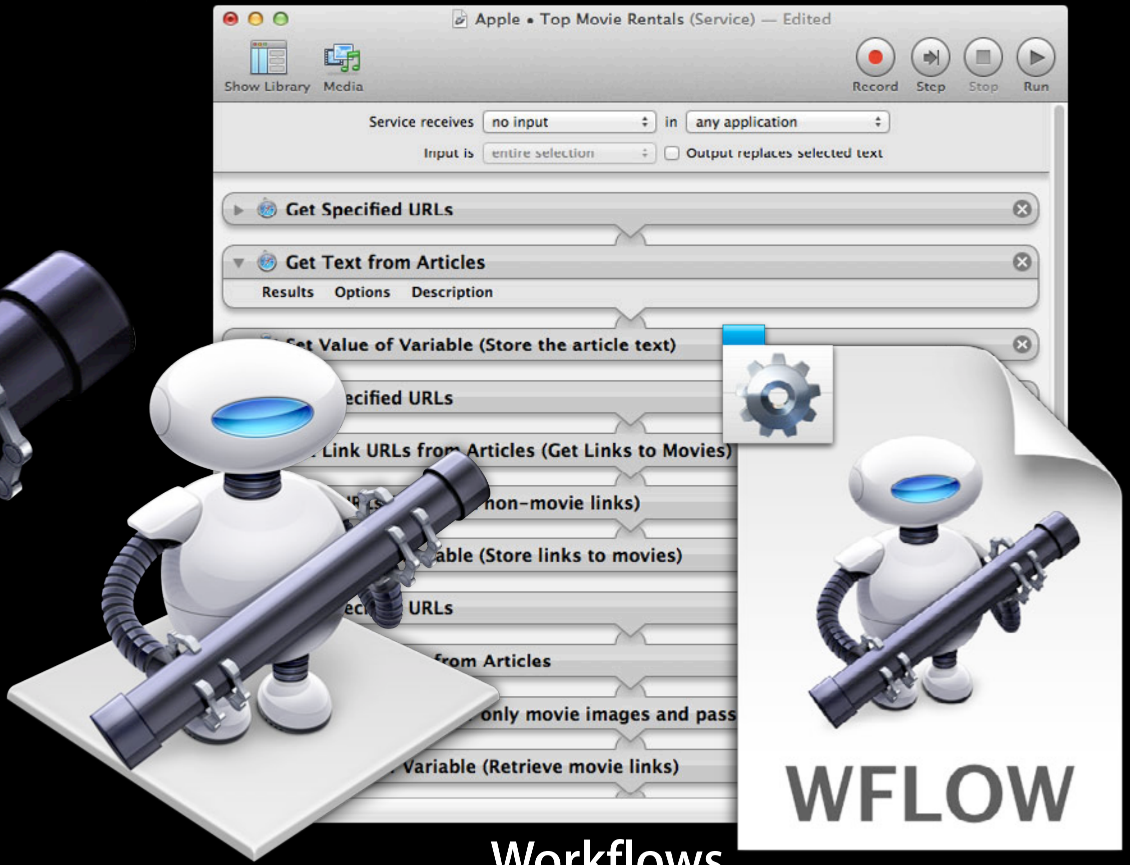


Applets

Workflows

Workflows, Scripts, and Applets

Automator



Service receives no input in any application

Input is entire selection Output replaces selected text

Set Value of Variable (Store the article text)

Link URLs from Articles (Get Links to Movies)

non-movie links)

able (Store links to movies)

URLs

from Articles

only movie images and pass

Variable (Retrieve movie links)

Applets

Workflows

Services

Workflows, Scripts, and Applets

Automator

Applets

Workflows

Services

Workflows, Scripts, and Applets

AppleScript Editor and System Script Menu

Workflows, Scripts, and Applets

AppleScript Editor and System Script Menu

```
tell application "Notes"
	set the folderCount to the
	if folderCount is 0 then
		display dialog "There
			{"Cancel"} d
	else
		display dialog "This scr
			with icon 1
	end if
	try
		set thisAccountName to t
		tell account thisAccountNa
			delete (every folder w
		end tell
	on error errorMessage number e
		display alert "ERROR " & er
			Error Text", "Canc
		tell application "TextEdit"
			ate
			ke new docu       with

	en
end tell
```

Workflows, Scripts, and Applets

AppleScript Editor and System Script Menu

```
tell application "Notes"
	set the folderCount to the
	if folderCount is 0 then
		display dialog "There
			{"Cancel"} d
	else
		display
	
	end if
	try
		set this
		tell acc
			de
		end te
	on error errorMessage number e
		display alert "ERROR " & er
			Error Text", "Canc
		tell application "TextEdit"
			ate
			ke new docu      with
	
	en
end tell
```

UNIX Commands

Running automation tools from the command line

- Shell scripts

```
bash
python
ruby
perl
osascript
...
```

UNIX Commands

Running automation tools from the command line

- Shell scripts

```
bash
python
ruby
perl
osascript
...
```

```
osascript -e 'tell application "TextEdit" to make new document'
automator -i ~/Pictures/DCS52.jpg ~/Library/Workflows/scale.workflow
```

UNIX Commands

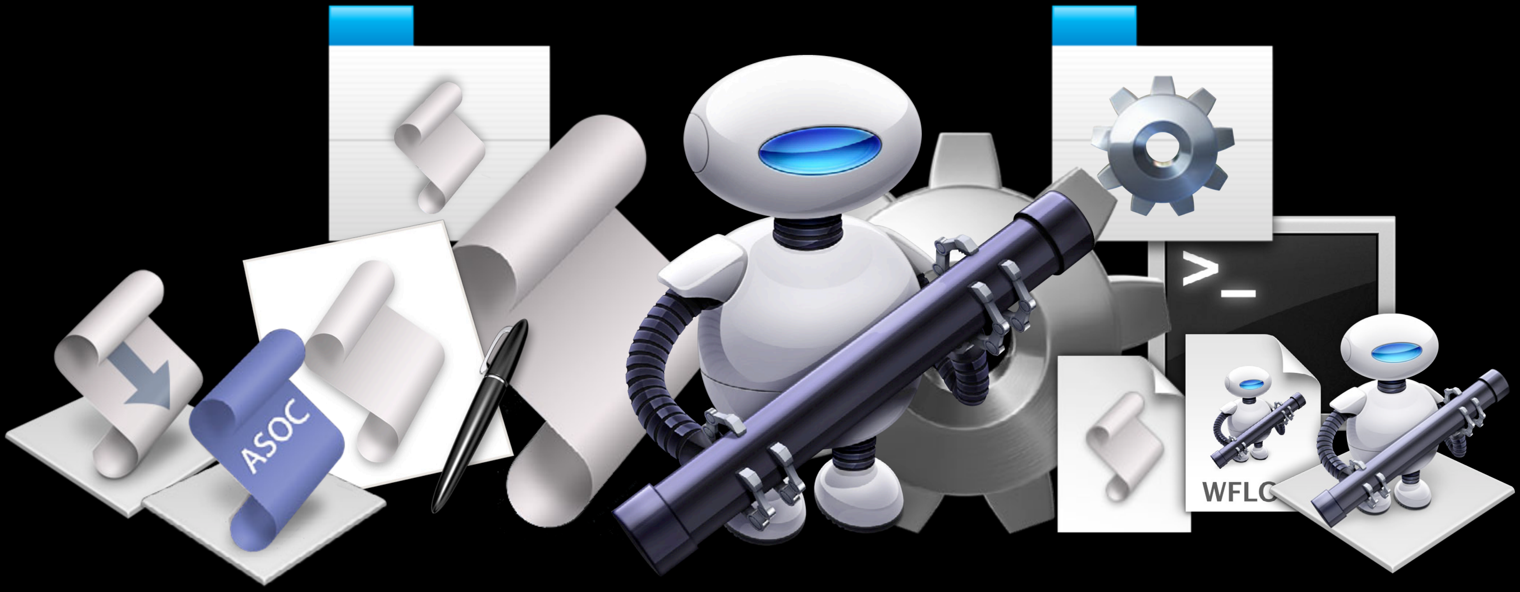
Running automation tools from the command line

- Shell scripts

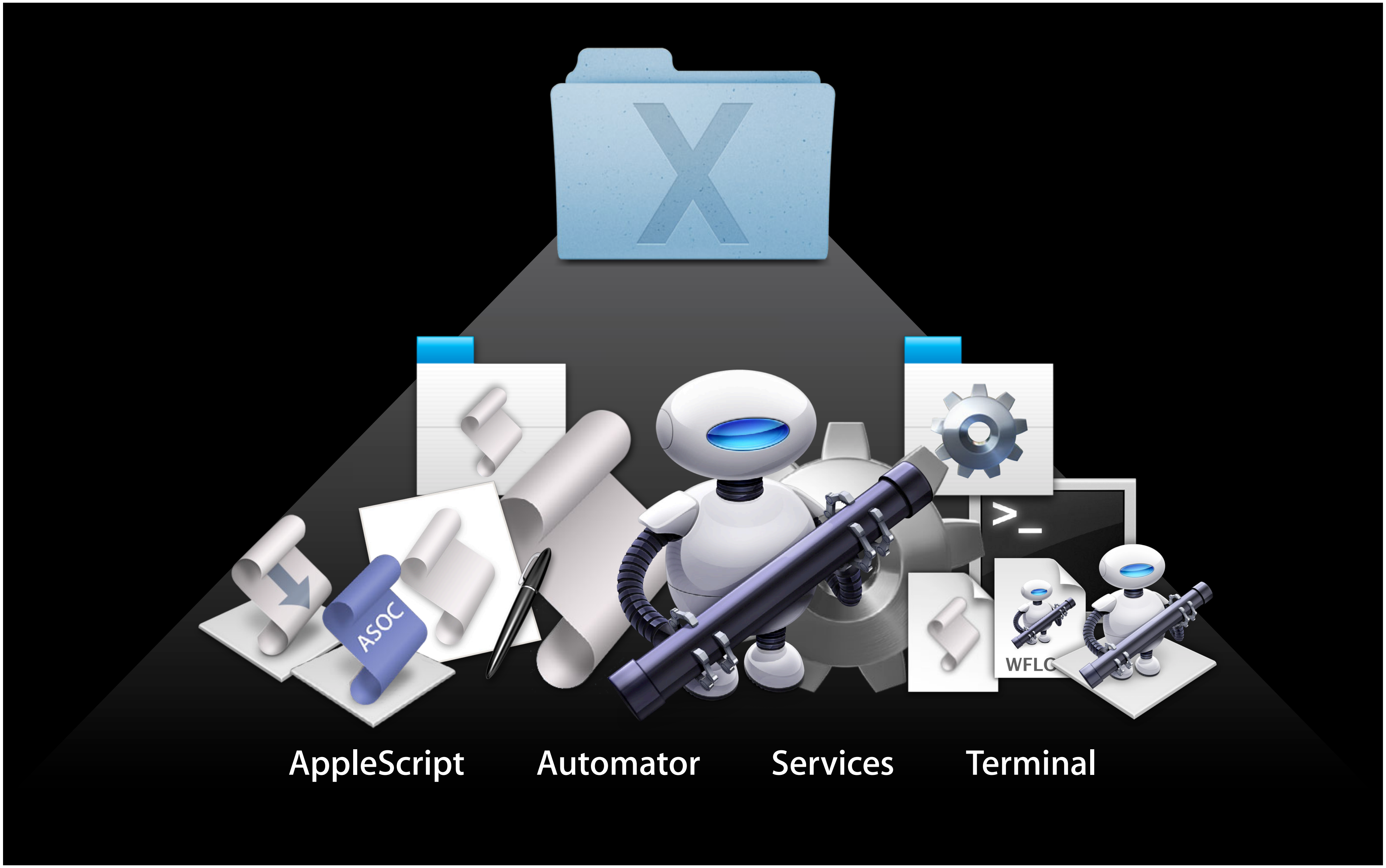
```
bash
python
ruby
perl
osascript
...
```

NO RESTRICTIONS ✓

```
osascript -e 'tell application "TextEdit" to make new document'
automator -i ~/Pictures/DCS52.jpg ~/Library/Workflows/scale.workflow
```



AppleScript Automator Services Terminal


ASOC
>_
WFLC
AppleScript
Automator
Services
Terminal

NO RESTRICTIONS ✓
ASOC
WFLC
AppleScript
Automator
Services
Terminal

Personal Automation

Scripts you write for yourself

- Scripts and workflows executed by the system are not restricted
  - Automator, AppleScript Editor, Script Menu, Services, Terminal

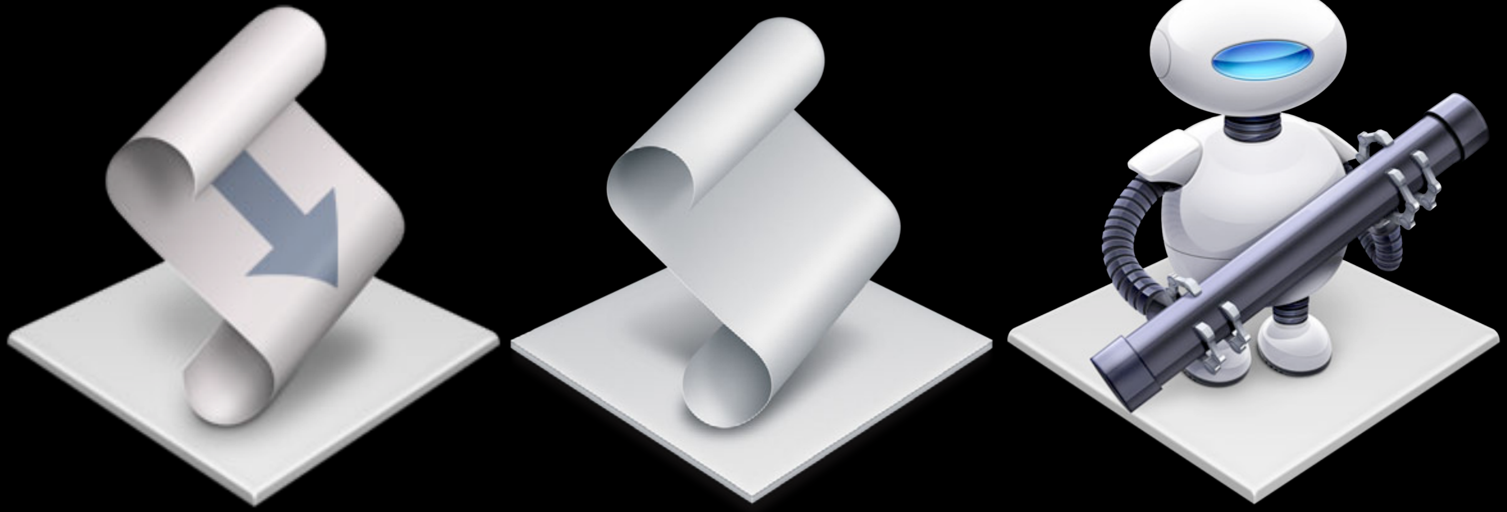
Personal Automation

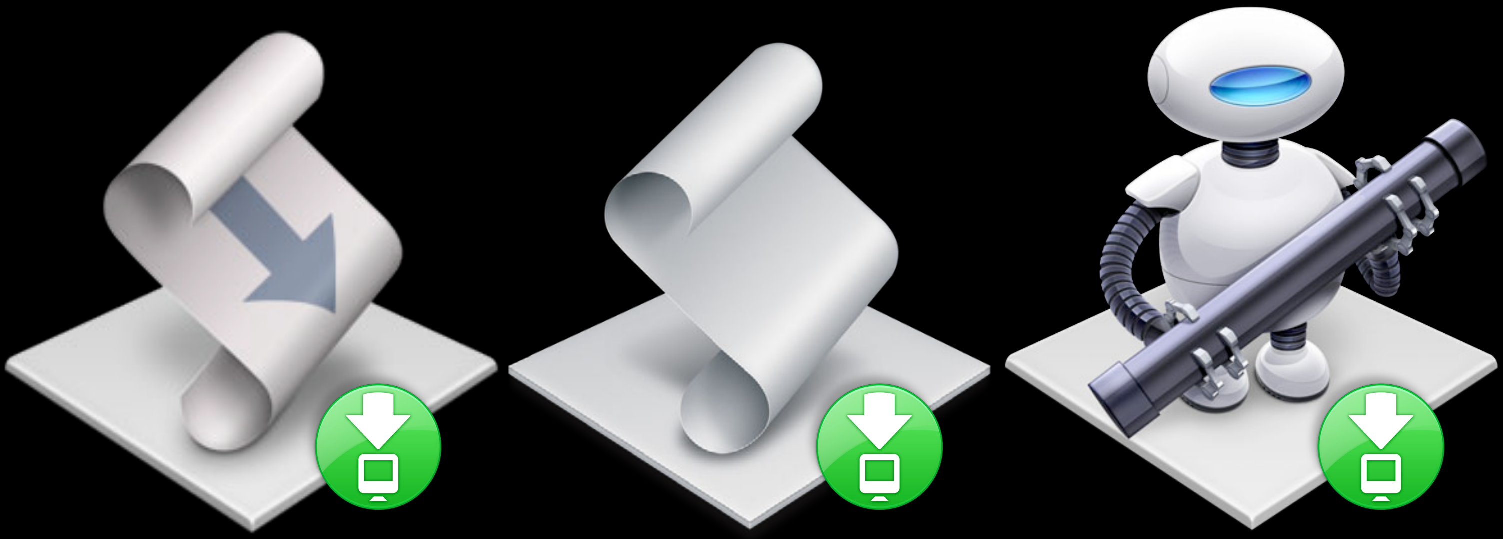
Scripts you write for yourself

- Scripts and workflows executed by the system are not restricted
  - Automator, AppleScript Editor, Script Menu, Services, Terminal

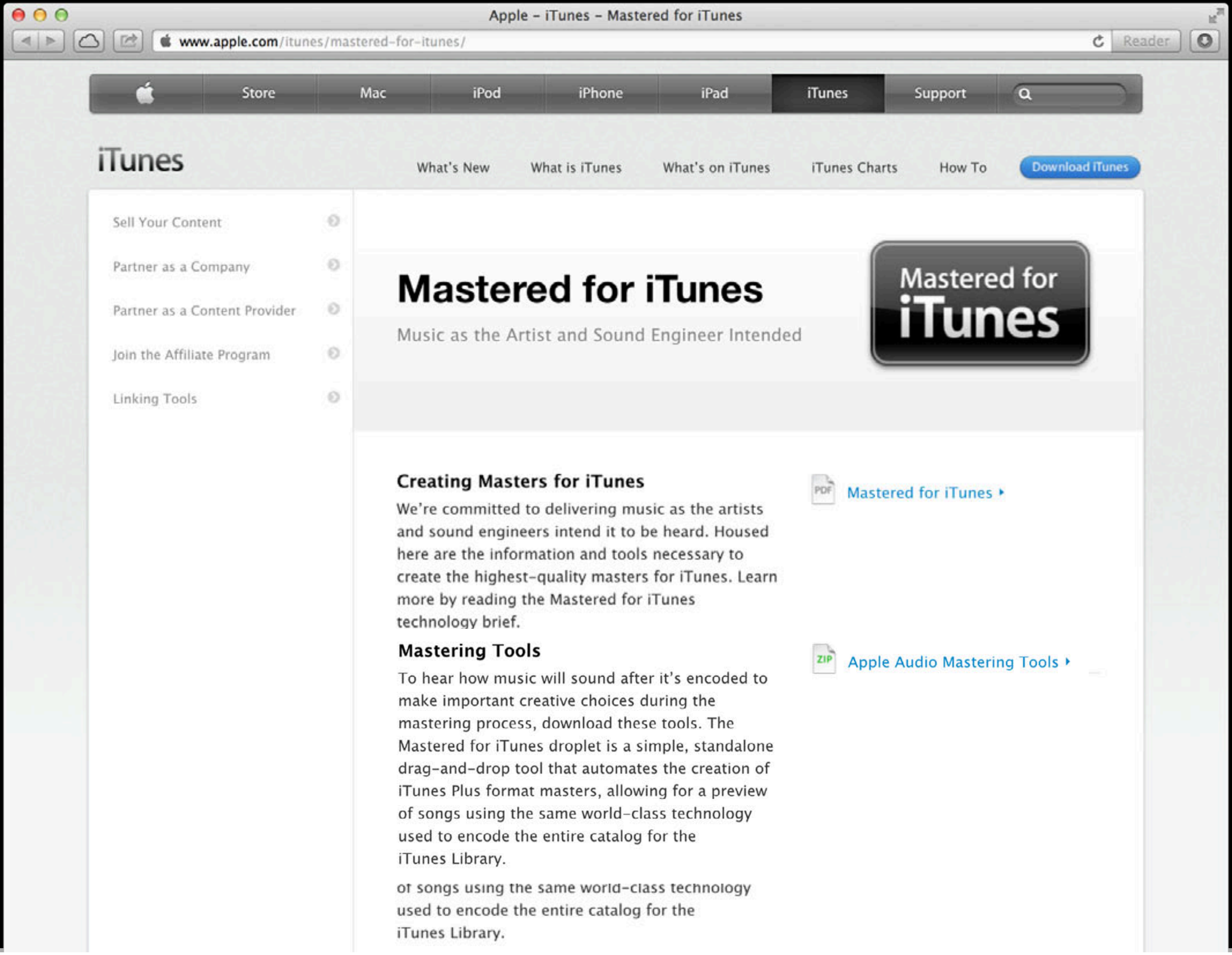
Distributing Scripts

Working with Gatekeeper





Apple Audio Mastering Tools



Mastered for iTunes

Music as the Artist and Sound Engineer Intended

Creating Masters for iTunes

We're committed to delivering music as the artists and sound engineers intend it to be heard. Housed here are the information and tools necessary to create the highest-quality masters for iTunes. Learn more by reading the Mastered for iTunes technology brief.

Mastered for iTunes ▸

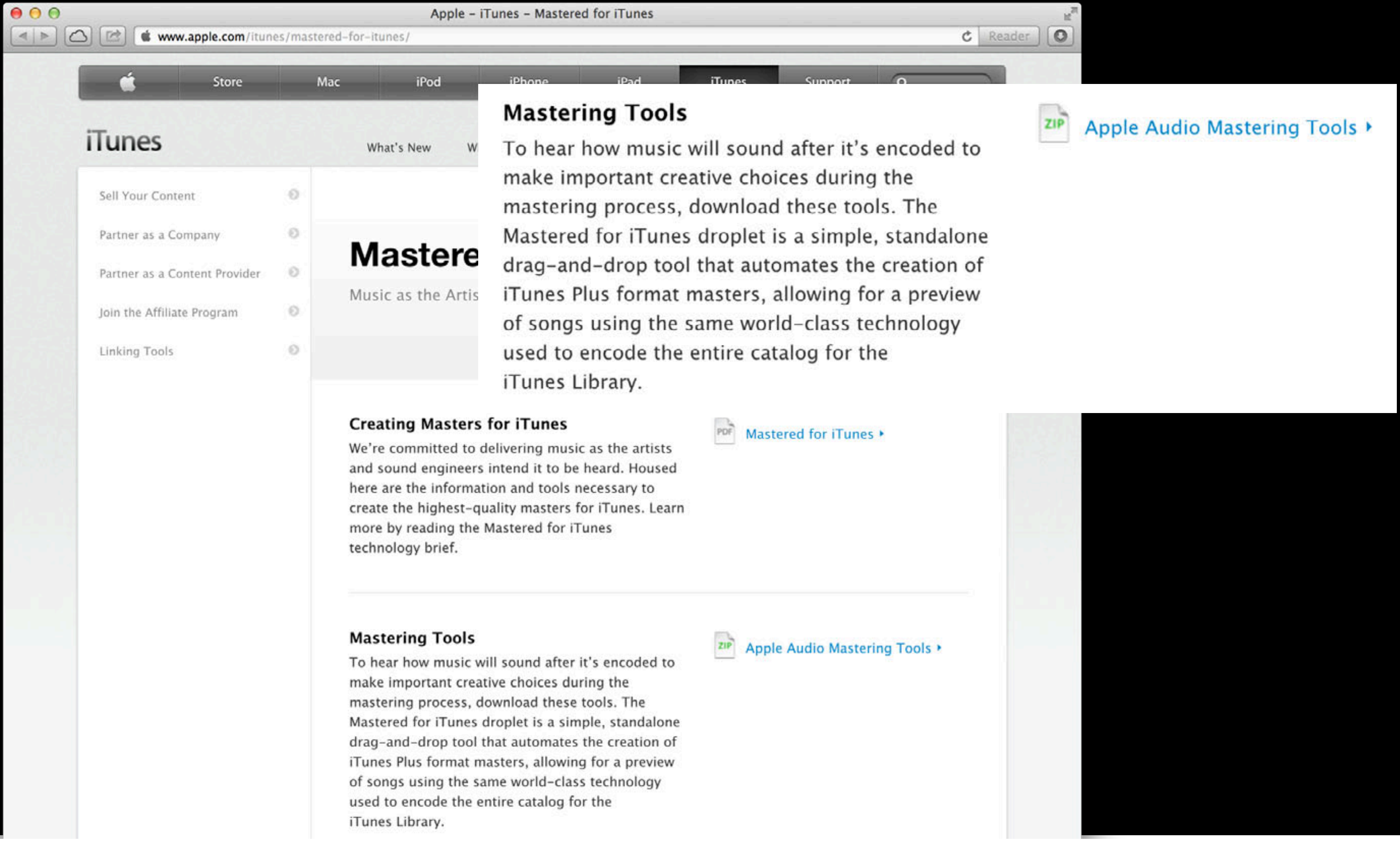
Mastering Tools

To hear how music will sound after it's encoded to make important creative choices during the mastering process, download these tools. The Mastered for iTunes droplet is a simple, standalone drag-and-drop tool that automates the creation of iTunes Plus format masters, allowing for a preview of songs using the same world-class technology used to encode the entire catalog for the iTunes Library.

Apple Audio Mastering Tools ▸

of songs using the same world-class technology used to encode the entire catalog for the iTunes Library.

Apple Audio Mastering Tools


Apple – iTunes – Mastered for iTunes
www.apple.com/itunes/mastered-for-itunes/
Reader
Store
Mac
iPod
iPhone
iPad
iTunes
Support
iTunes
What's New
Sell Your Content
Partner as a Company
Partner as a Content Provider
Join the Affiliate Program
Linking Tools
Mastere
Music as the Artis

Mastering Tools

To hear how music will sound after it's encoded to make important creative choices during the mastering process, download these tools. The Mastered for iTunes droplet is a simple, standalone drag-and-drop tool that automates the creation of iTunes Plus format masters, allowing for a preview of songs using the same world-class technology used to encode the entire catalog for the iTunes Library.

Apple Audio Mastering Tools ▸

Creating Masters for iTunes

We're committed to delivering music as the artists and sound engineers intend it to be heard. Housed here are the information and tools necessary to create the highest-quality masters for iTunes. Learn more by reading the Mastered for iTunes technology brief.

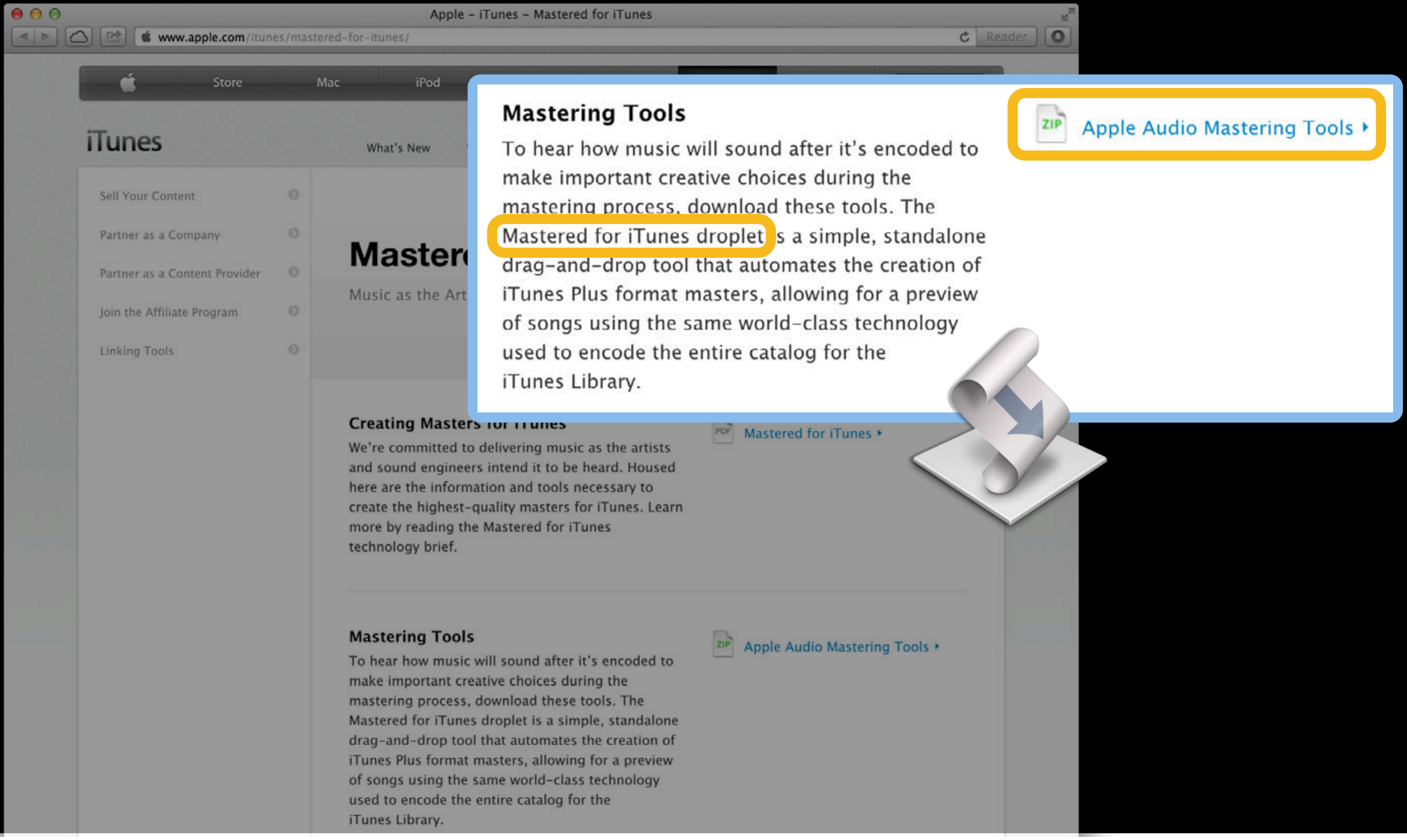
Mastered for iTunes ▸

Mastering Tools

To hear how music will sound after it's encoded to make important creative choices during the mastering process, download these tools. The Mastered for iTunes droplet is a simple, standalone drag-and-drop tool that automates the creation of iTunes Plus format masters, allowing for a preview of songs using the same world-class technology used to encode the entire catalog for the iTunes Library.

Apple Audio Mastering Tools ▸

Apple Audio Mastering Tools



Apple – iTunes – Mastered for iTunes
www.apple.com/itunes/mastered-for-itunes/
Reader
Store
Mac
iPod
iTunes
What's New
Sell Your Content
Partner as a Company
Partner as a Content Provider
Join the Affiliate Program
Linking Tools
Mastere
Music as the Art

Mastering Tools

To hear how music will sound after it's encoded to make important creative choices during the mastering process, download these tools. The Mastered for iTunes droplet is a simple, standalone drag-and-drop tool that automates the creation of iTunes Plus format masters, allowing for a preview of songs using the same world-class technology used to encode the entire catalog for the iTunes Library.

Apple Audio Mastering Tools ›

Creating Masters for iTunes

We're committed to delivering music as the artists and sound engineers intend it to be heard. Housed here are the information and tools necessary to create the highest-quality masters for iTunes. Learn more by reading the Mastered for iTunes technology brief.

Mastered for iTunes ›

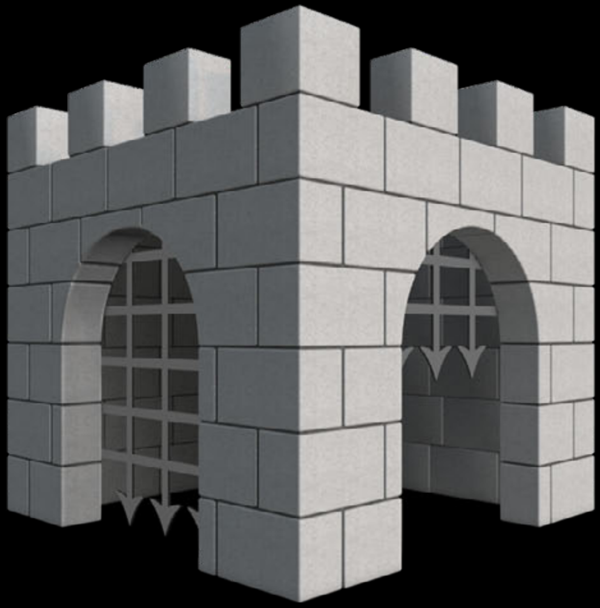
Mastering Tools

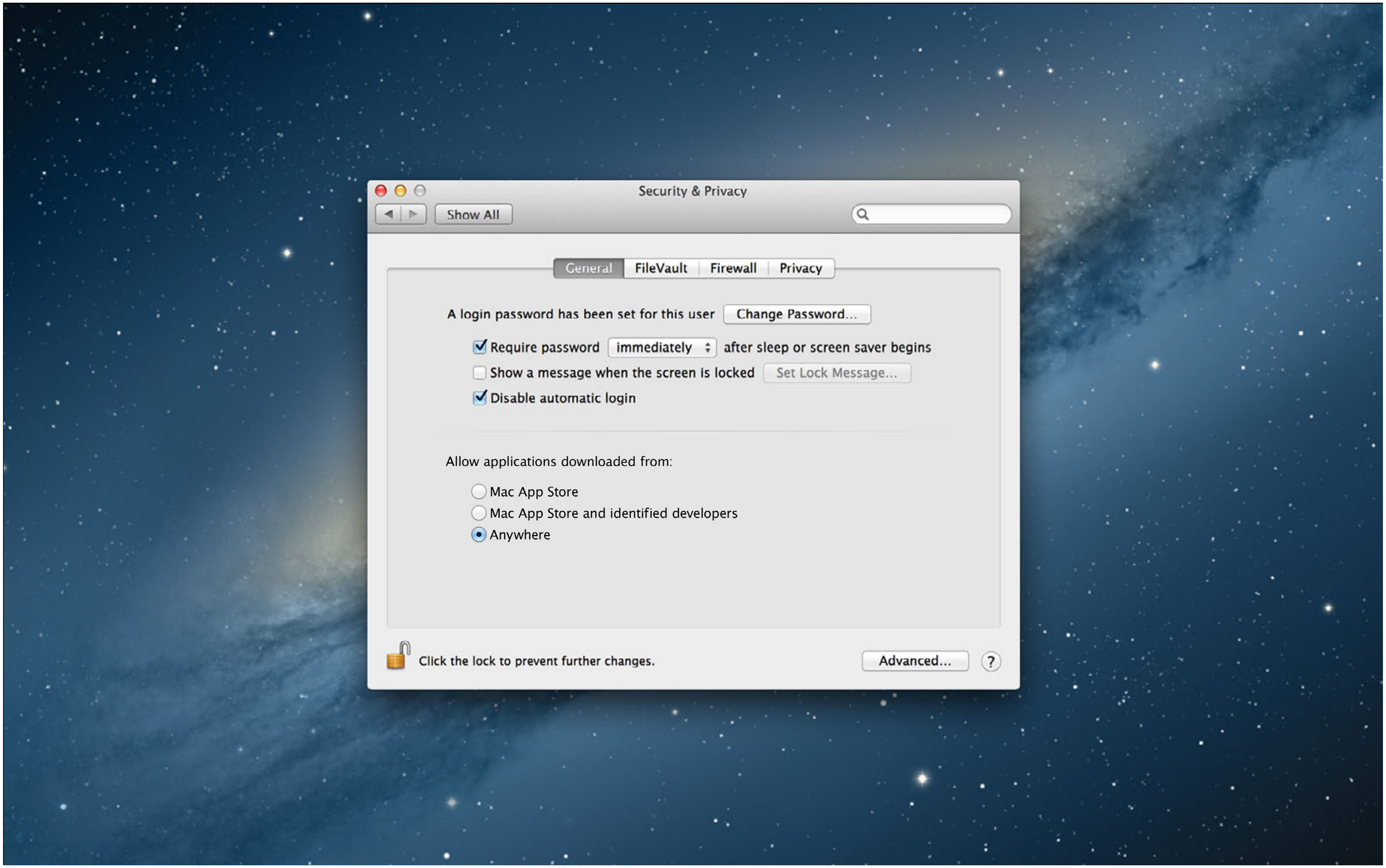
To hear how music will sound after it's encoded to make important creative choices during the mastering process, download these tools. The Mastered for iTunes droplet is a simple, standalone drag-and-drop tool that automates the creation of iTunes Plus format masters, allowing for a preview of songs using the same world-class technology used to encode the entire catalog for the iTunes Library.

Apple Audio Mastering Tools ›

Gatekeeper

Makes it safer to download software





A login password has been set for this user [Change Password...]

[✓] Require password [immediately] after sleep or screen saver begins

[ ] Show a message when the screen is locked [Set Lock Message...]

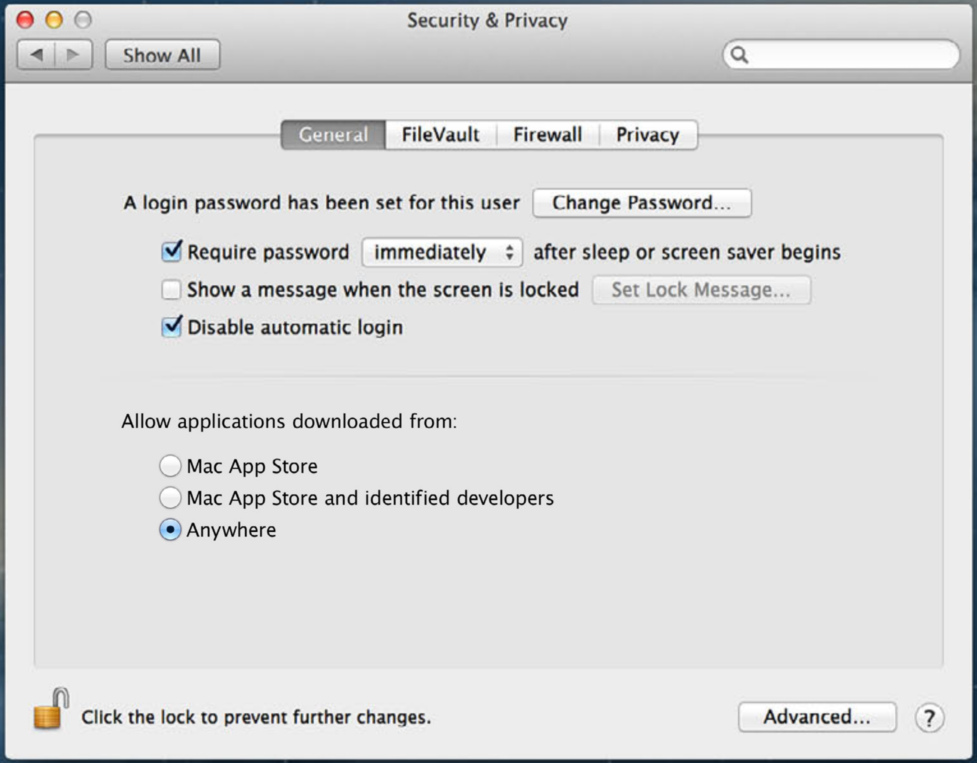
[✓] Disable automatic login

Allow applications downloaded from:

- ( ) Mac App Store
- ( ) Mac App Store and identified developers
- [✓ Anywhere]

Click the lock to prevent further changes.

Security & Privacy Preferences


Security & Privacy

A login password has been set for this user [Change Password…]

[✓] Require password [immediately] after sleep or screen saver begins

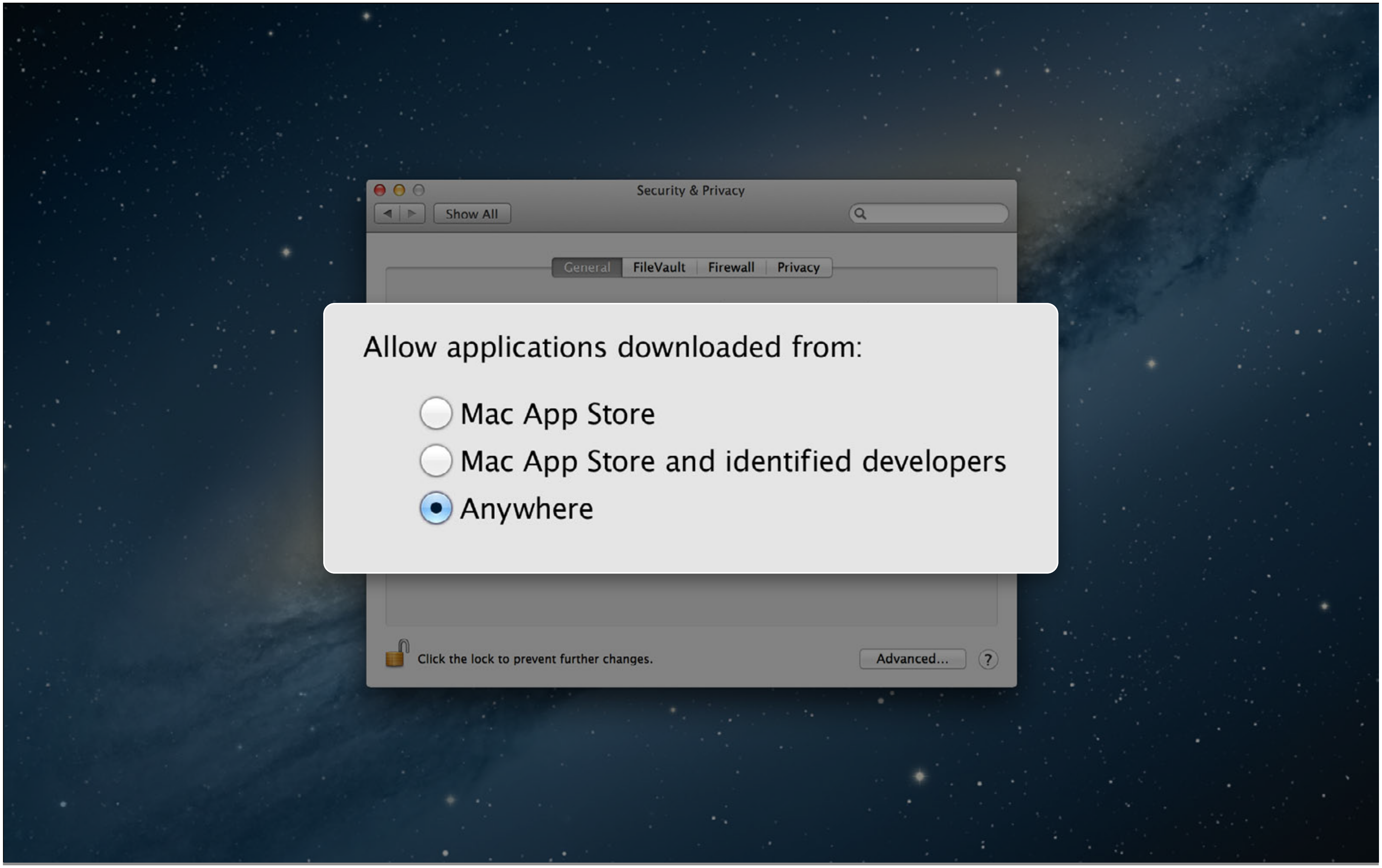
[ ] Show a message when the screen is locked [Set Lock Message…]

[✓] Disable automatic login

Allow applications downloaded from:

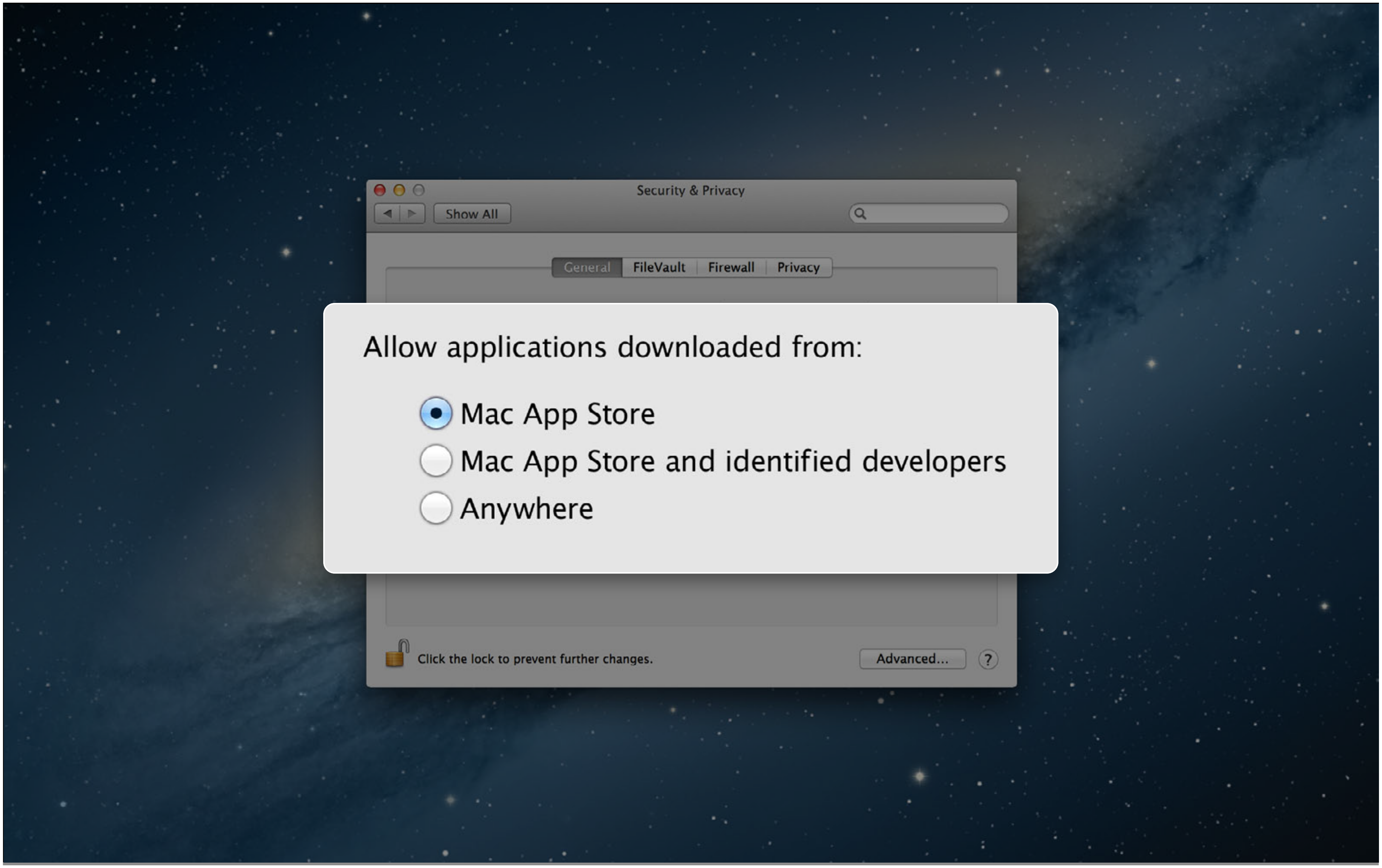
- ○ Mac App Store
- ○ Mac App Store and identified developers
- ● Anywhere

Click the lock to prevent further changes.



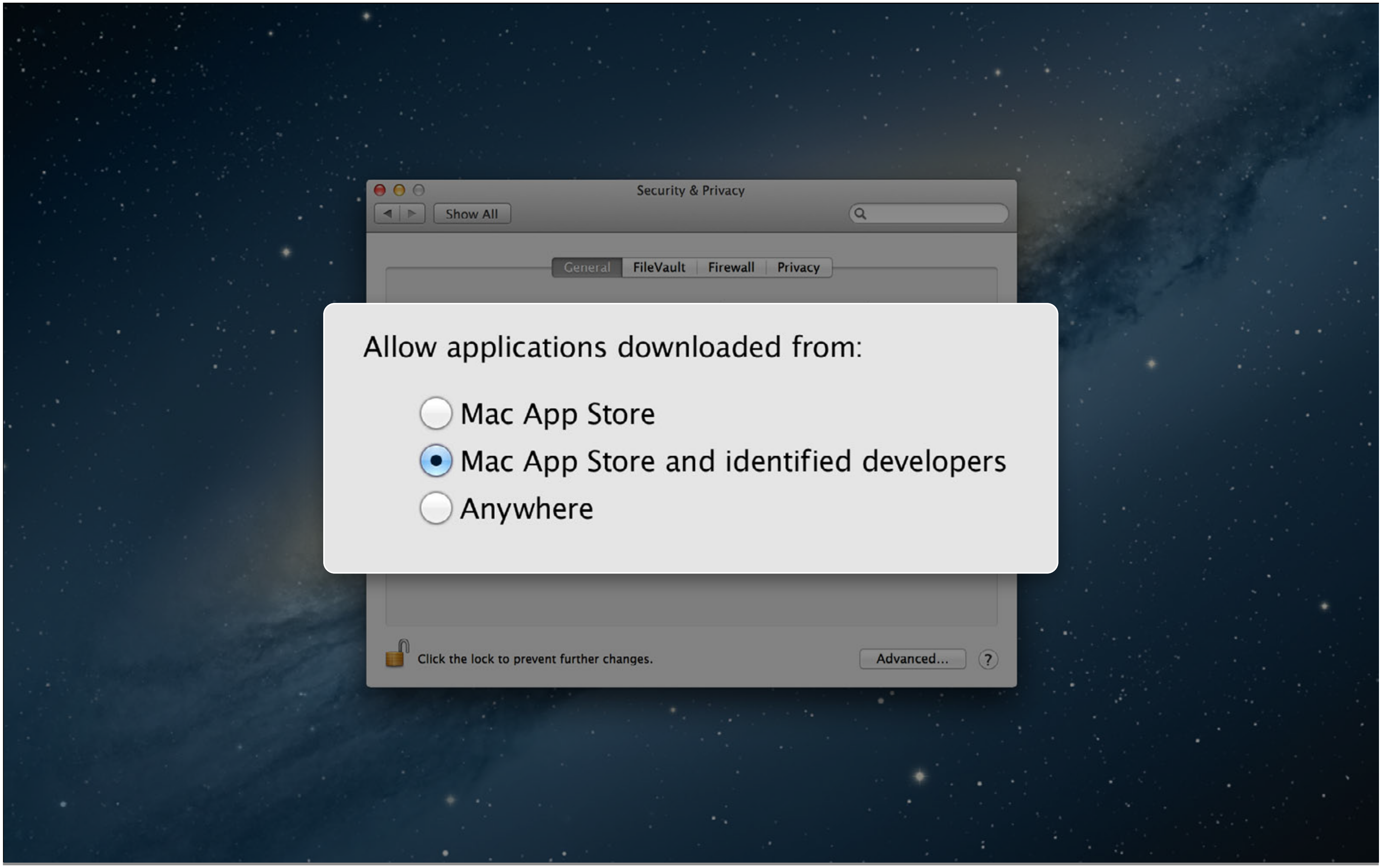
Allow applications downloaded from:

- Mac App Store
- Mac App Store and identified developers
- [✓ Anywhere]



Allow applications downloaded from:

[✓ Mac App Store]
Mac App Store and identified developers
Anywhere

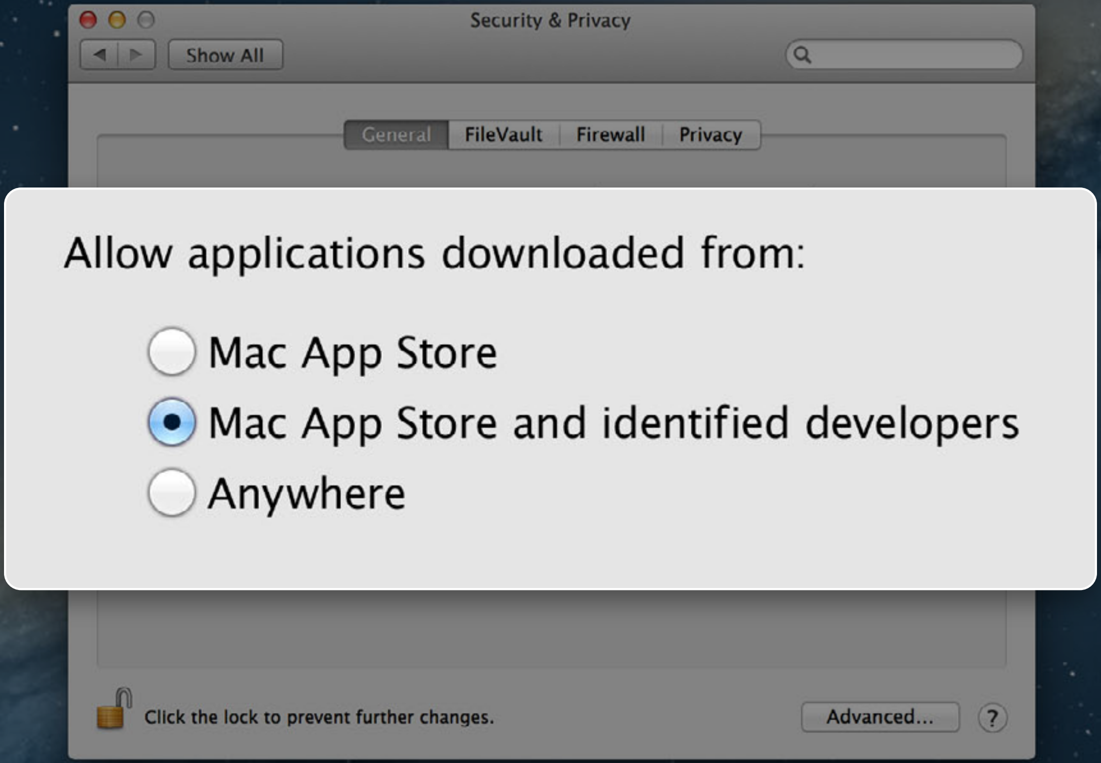


Allow applications downloaded from:

- ( ) Mac App Store
- (●) Mac App Store and identified developers
- ( ) Anywhere

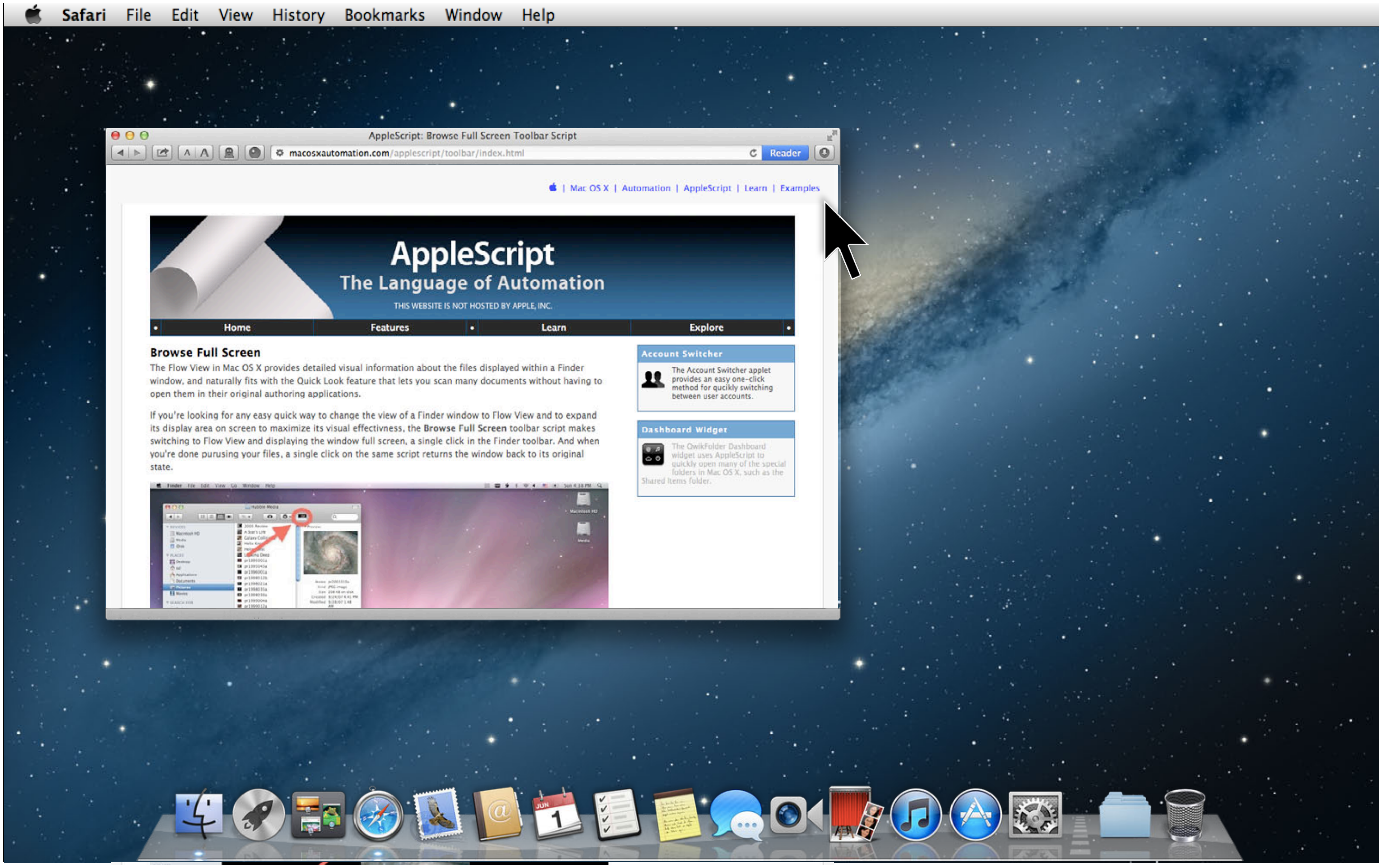
Click the lock to prevent further changes.

Default Setting



Allow applications downloaded from:

- [ ] Mac App Store
- [✓ Mac App Store and identified developers]
- [ ] Anywhere


Safari File Edit View History Bookmarks Window Help
AppleScript: Browse Full Screen Toolbar Script
macosxautomation.com/applescript/toolbar/index.html
Reader
| Mac OS X | Automation | AppleScript | Learn | Examples
AppleScript
The Language of Automation
THIS WEBSITE IS NOT HOSTED BY APPLE, INC.
Home
Features
Learn
Explore

Browse Full Screen

The Flow View in Mac OS X provides detailed visual information about the files displayed within a Finder window, and naturally fits with the Quick Look feature that lets you scan many documents without having to open them in their original authoring applications.

If you're looking for any easy quick way to change the view of a Finder window to Flow View and to expand its display area on screen to maximize its visual effectivness, the **Browse Full Screen** toolbar script makes switching to Flow View and displaying the window full screen, a single click in the Finder toolbar. And when you're done purusing your files, a single click on the same script returns the window back to its original state.

Account Switcher

The Account Switcher applet provides an easy one-click method for qucikly switching between user accounts.

Dashboard Widget

The QwikFolder Dashboard widget uses AppleScript to quickly open many of the special folders in Mac OS X, such as the Shared Items folder.

Safari File Edit View History Bookmarks Window Help
AppleScript: Browse Full Screen Toolbar Script
macosxautomation.com/applescript/toolbar/index.html
Reader
Mac OS X | Automation | AppleScript | Learn | Examples
Home
Features
Learn
Explore

macosxautomation.com

AppleScript
The Language of Automation
THIS WEBSITE IS NOT HOSTED BY APPLE, INC.

Browse Full Screen

The Flow View in Mac OS X provides detailed visual information about the files displayed within a Finder window, and naturally fits with the Quick Look feature that lets you scan many documents without having to open them in their original authoring applications.

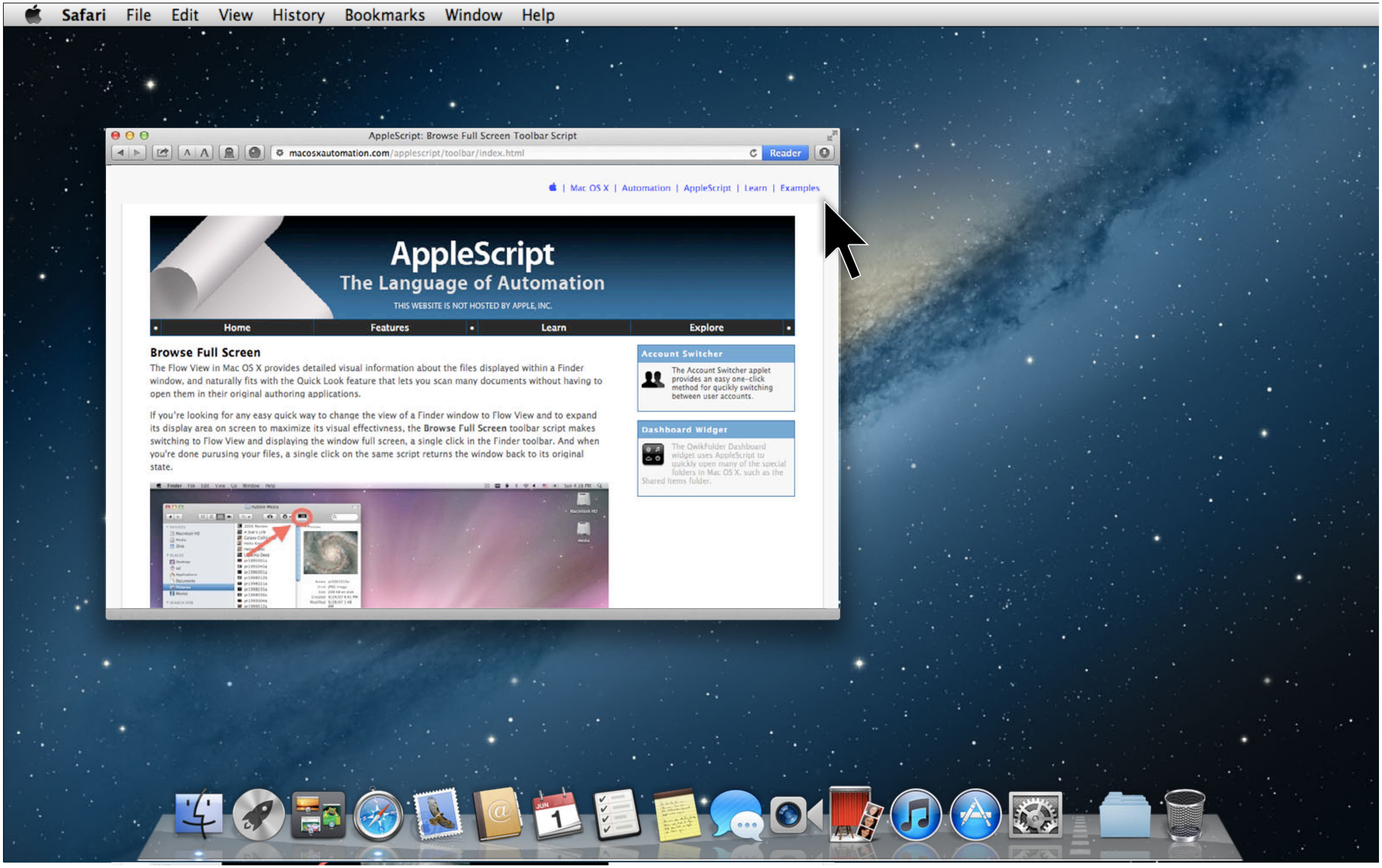
If you're looking for any easy quick way to change the view of a Finder window to Flow View and to expand its display area on screen to maximize its visual effectivness, the **Browse Full Screen** toolbar script makes switching to Flow View and displaying the window full screen, a single click in the Finder toolbar. And when you're done purusing your files, a single click on the same script returns the window back to its original state.

Account Switcher

The Account Switcher applet provides an easy one-click method for qucikly switching between user accounts.

Dashboard Widget

The QwikFolder Dashboard widget uses AppleScript to quickly open many of the special folders in Mac OS X, such as the Shared Items folder.


Safari
File
Edit
View
History
Bookmarks
Window
Help
AppleScript: Browse Full Screen Toolbar Script
macosxautomation.com/applescript/toolbar/index.html
Reader
| Mac OS X | Automation | AppleScript | Learn | Examples
AppleScript
The Language of Automation
THIS WEBSITE IS NOT HOSTED BY APPLE, INC.
Home
Features
Learn
Explore

Browse Full Screen

The Flow View in Mac OS X provides detailed visual information about the files displayed within a Finder window, and naturally fits with the Quick Look feature that lets you scan many documents without having to open them in their original authoring applications.

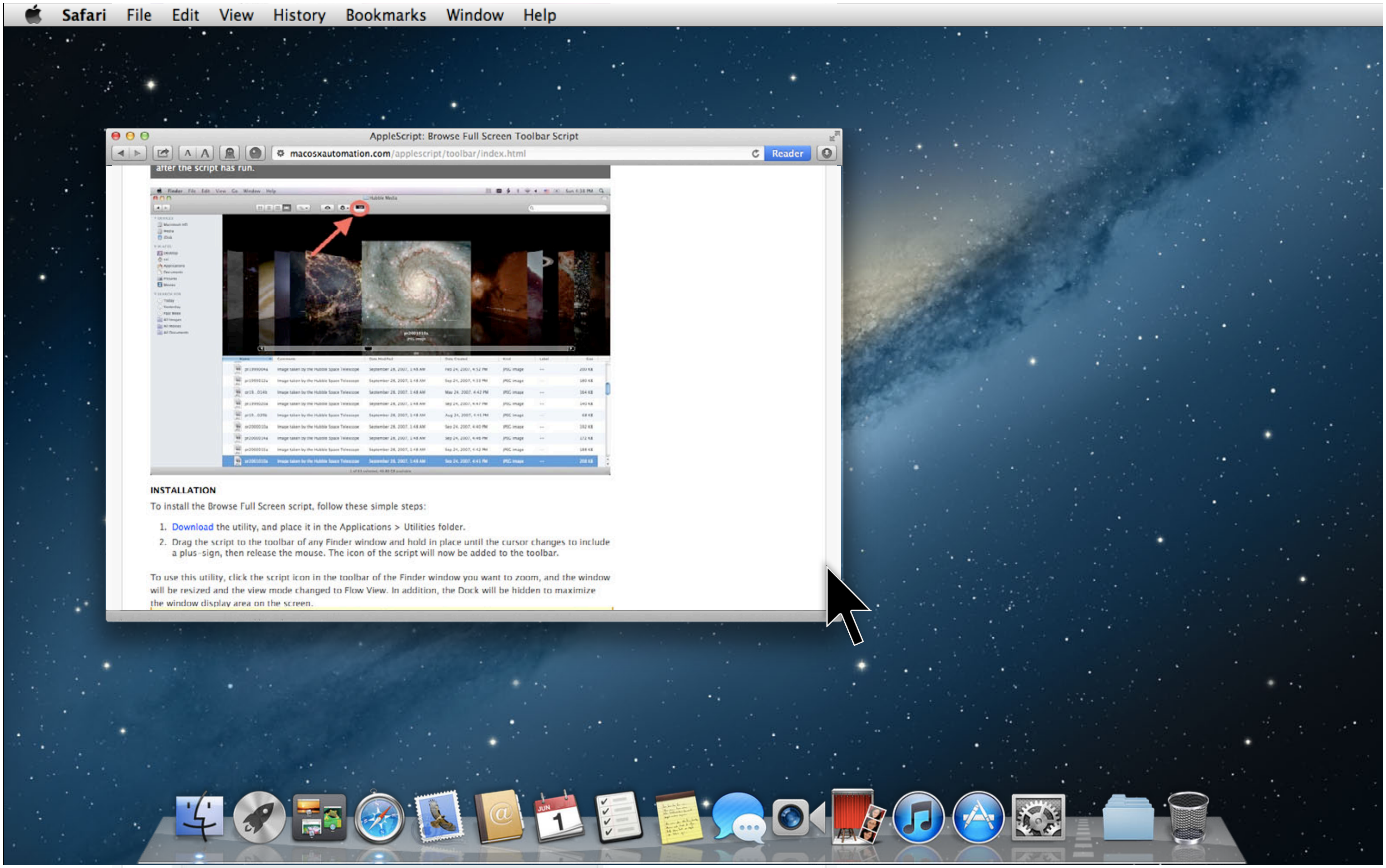
If you're looking for any easy quick way to change the view of a Finder window to Flow View and to expand its display area on screen to maximize its visual effectivness, the **Browse Full Screen** toolbar script makes switching to Flow View and displaying the window full screen, a single click in the Finder toolbar. And when you're done purusing your files, a single click on the same script returns the window back to its original state.

Account Switcher

The Account Switcher applet provides an easy one-click method for qucikly switching between user accounts.

Dashboard Widget

The QwikFolder Dashboard widget uses AppleScript to quickly open many of the special folders in Mac OS X, such as the Shared Items folder.


Safari
File
Edit
View
History
Bookmarks
Window
Help
AppleScript: Browse Full Screen Toolbar Script
macosxautomation.com/applescript/toolbar/index.html
Reader

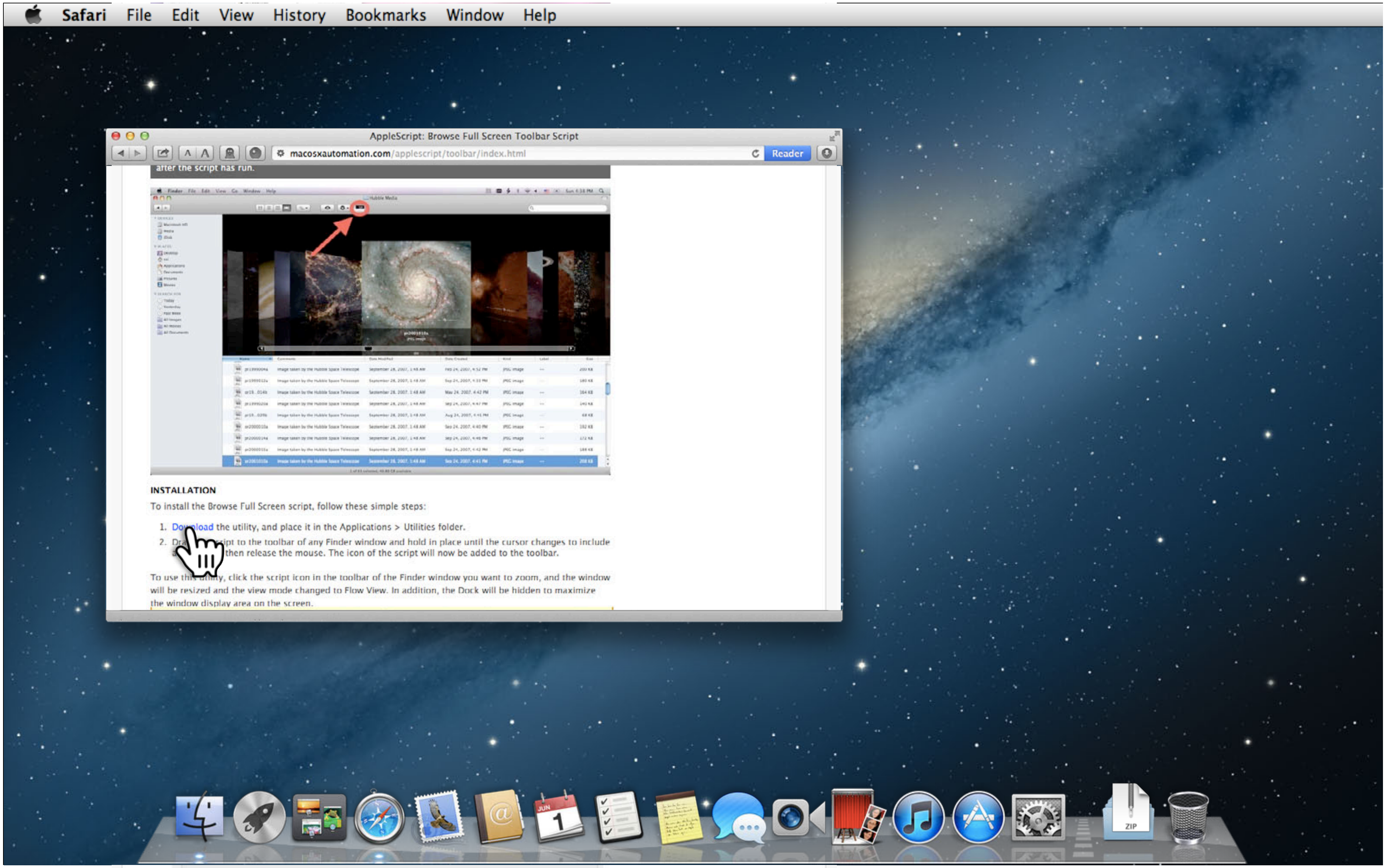
after the script has run.

INSTALLATION

To install the Browse Full Screen script, follow these simple steps:

1. Download the utility, and place it in the Applications > Utilities folder.
2. Drag the script to the toolbar of any Finder window and hold in place until the cursor changes to include a plus-sign, then release the mouse. The icon of the script will now be added to the toolbar.

To use this utility, click the script icon in the toolbar of the Finder window you want to zoom, and the window will be resized and the view mode changed to Flow View. In addition, the Dock will be hidden to maximize the window display area on the screen.


Safari
File
Edit
View
History
Bookmarks
Window
Help
AppleScript: Browse Full Screen Toolbar Script
macosxautomation.com/applescript/toolbar/index.html
Reader

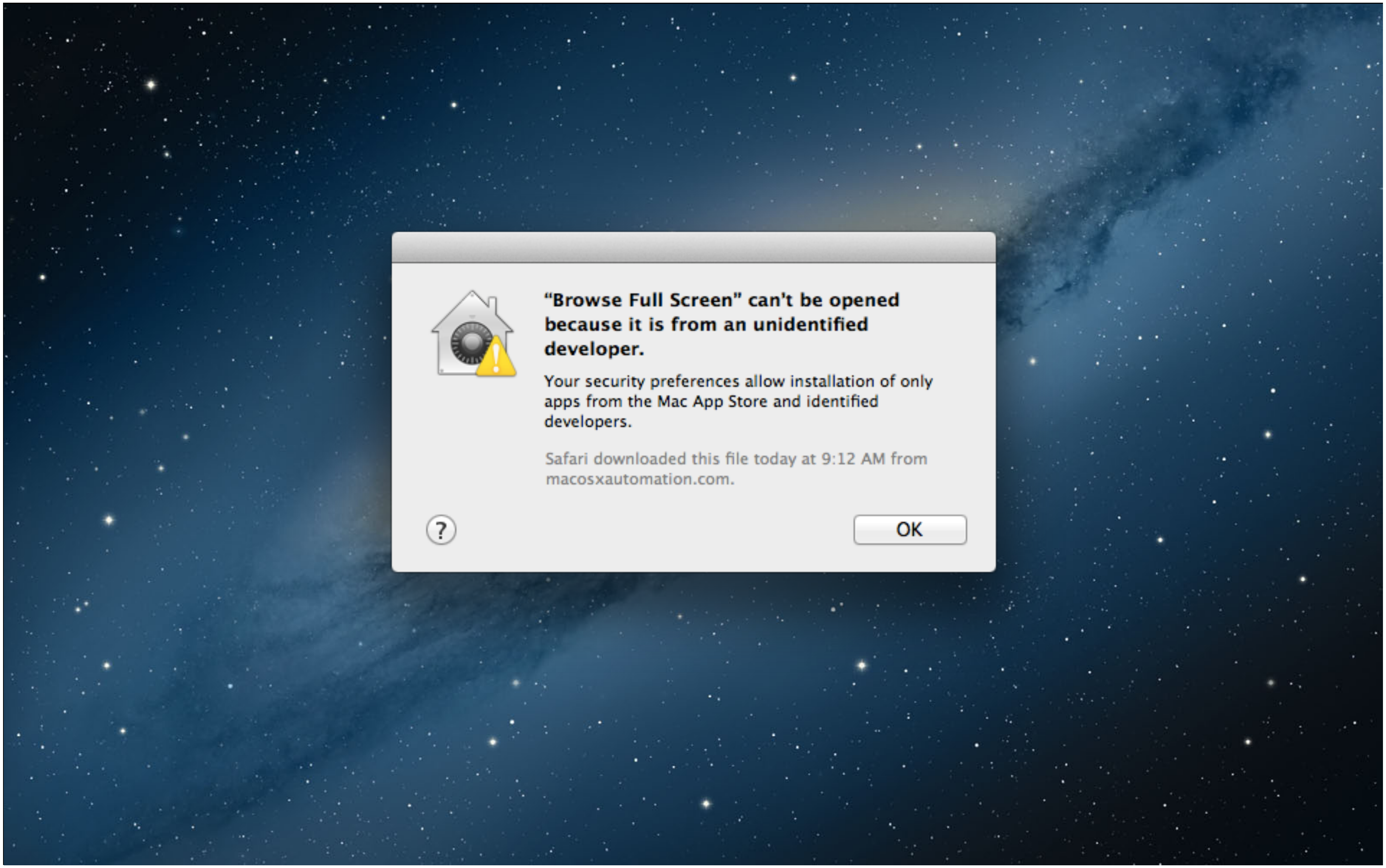
after the script has run.

INSTALLATION

To install the Browse Full Screen script, follow these simple steps:

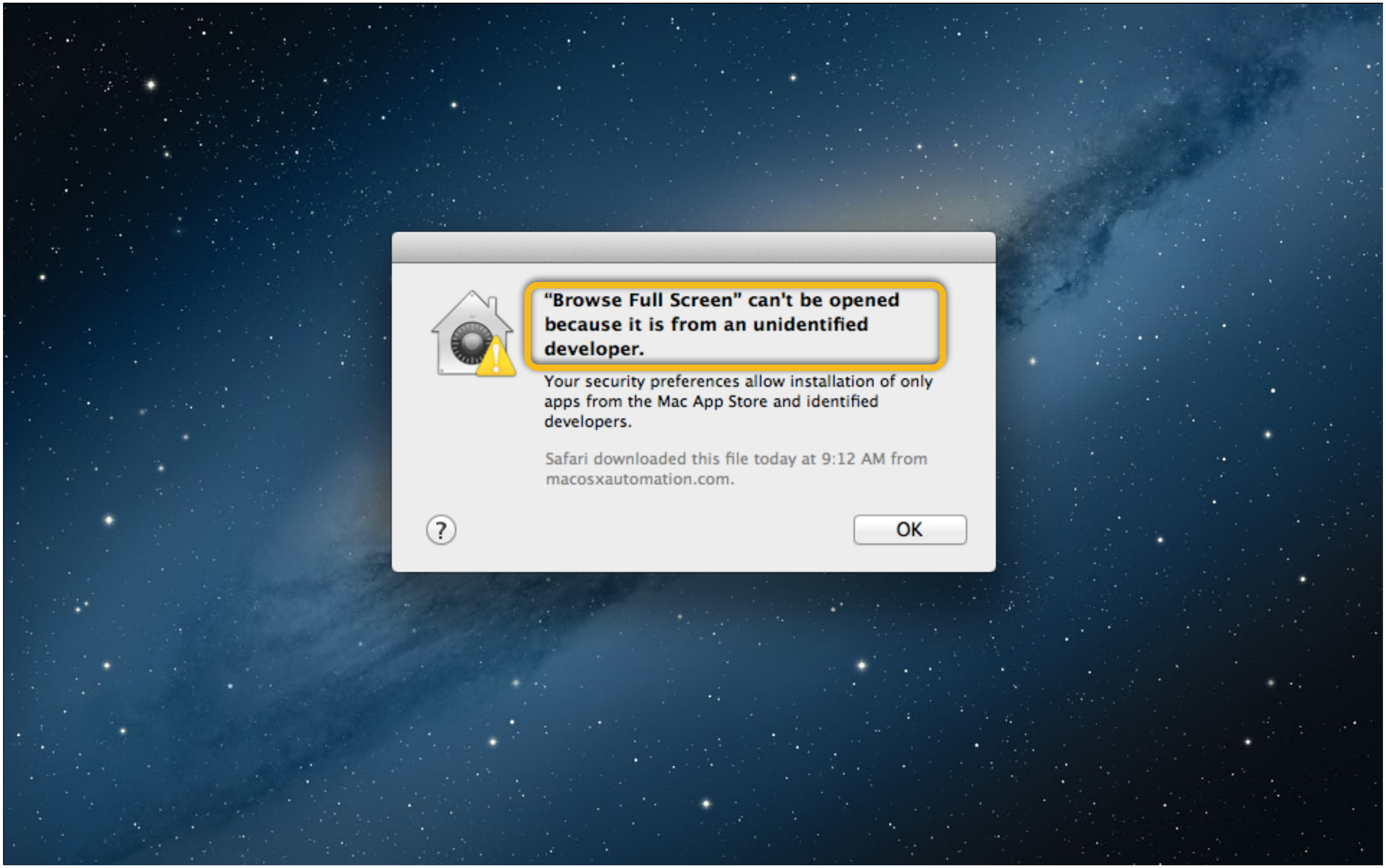
1. Download the utility, and place it in the Applications > Utilities folder.
2. Dra[illegible]ipt to the toolbar of any Finder window and hold in place until the cursor changes to include [illegible] then release the mouse. The icon of the script will now be added to the toolbar.

To use thi[illegible]y, click the script icon in the toolbar of the Finder window you want to zoom, and the window will be resized and the view mode changed to Flow View. In addition, the Dock will be hidden to maximize the window display area on the screen.


“Browse Full Screen” can’t be opened because it is from an unidentified developer.

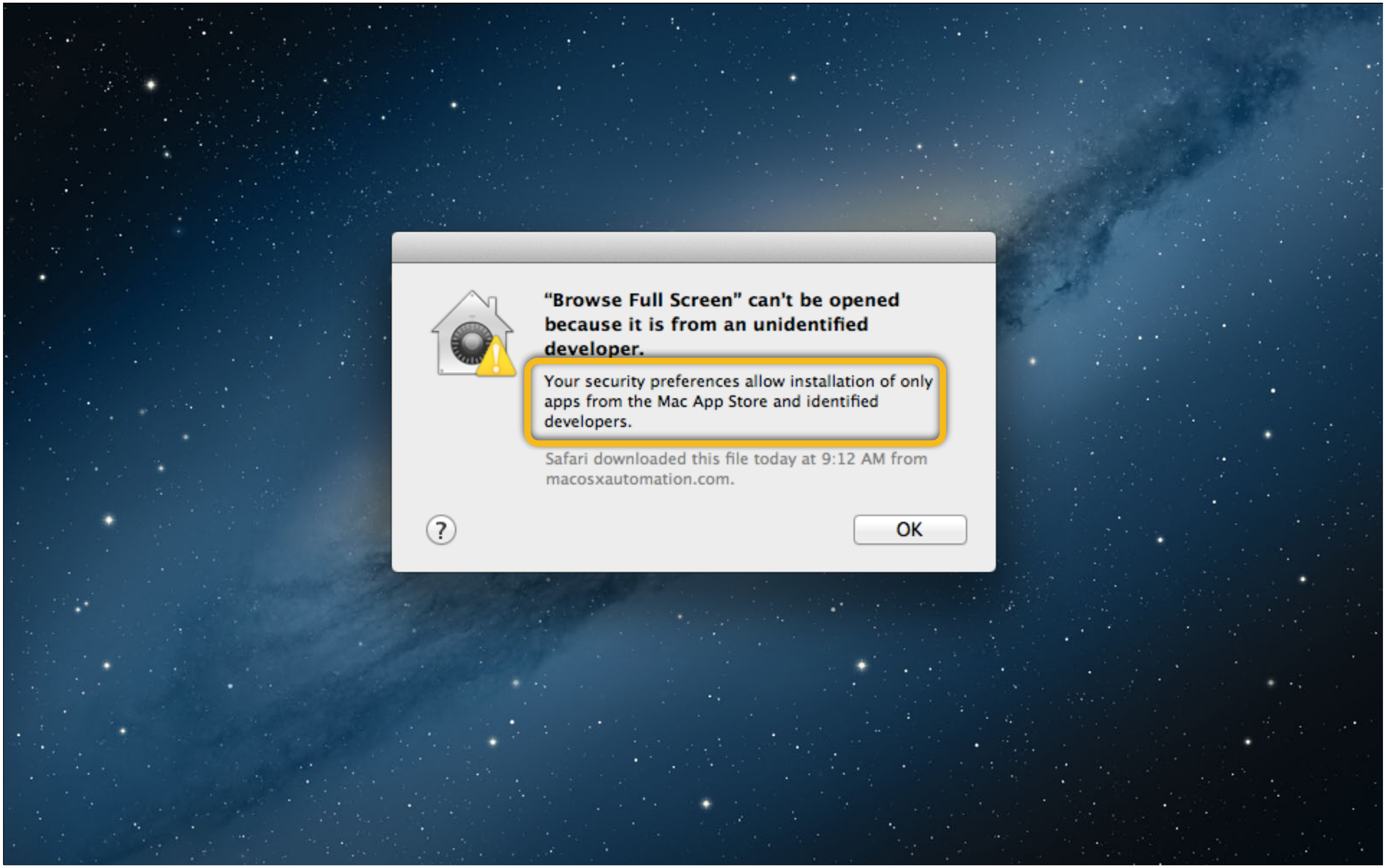
Your security preferences allow installation of only apps from the Mac App Store and identified developers.

Safari downloaded this file today at 9:12 AM from macosxautomation.com.


“Browse Full Screen” can't be opened because it is from an unidentified developer.

Your security preferences allow installation of only apps from the Mac App Store and identified developers.

Safari downloaded this file today at 9:12 AM from macosxautomation.com.


“Browse Full Screen” can’t be opened because it is from an unidentified developer.

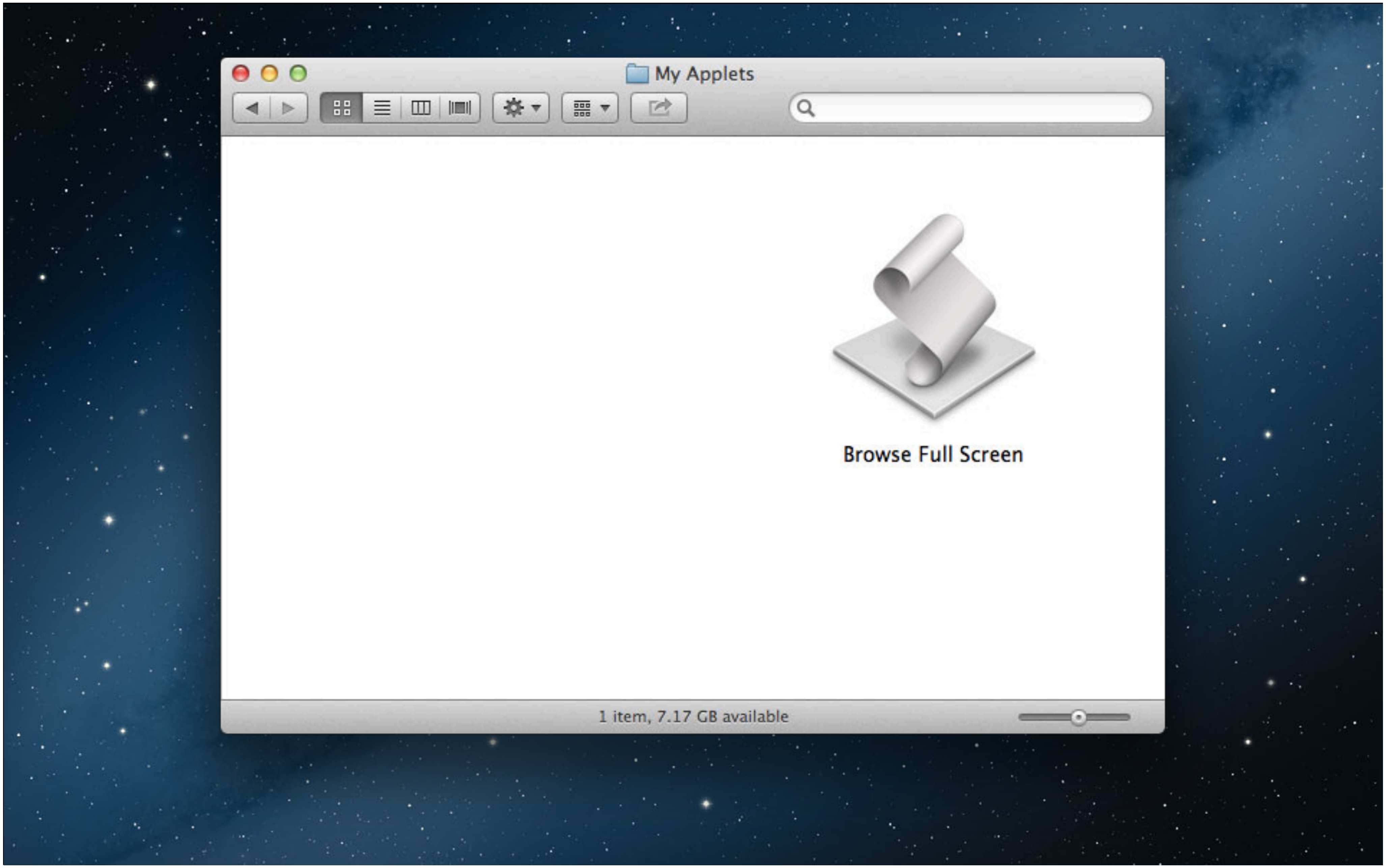
Your security preferences allow installation of only apps from the Mac App Store and identified developers.

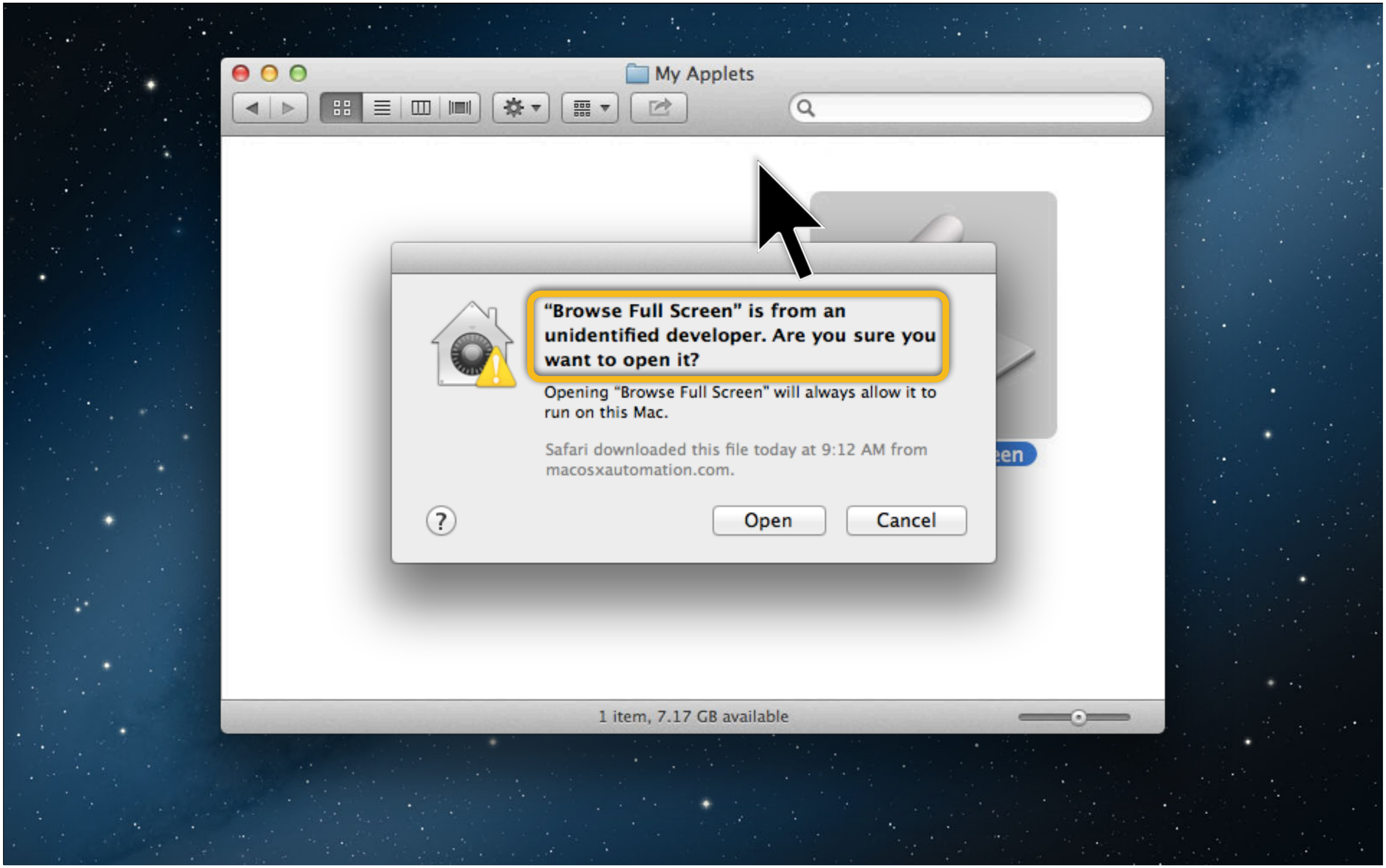
Safari downloaded this file today at 9:12 AM from macosxautomation.com.

ull Scre pened
it is from ed
develo er.

Your security pr llation of only
apps from th identified
developers

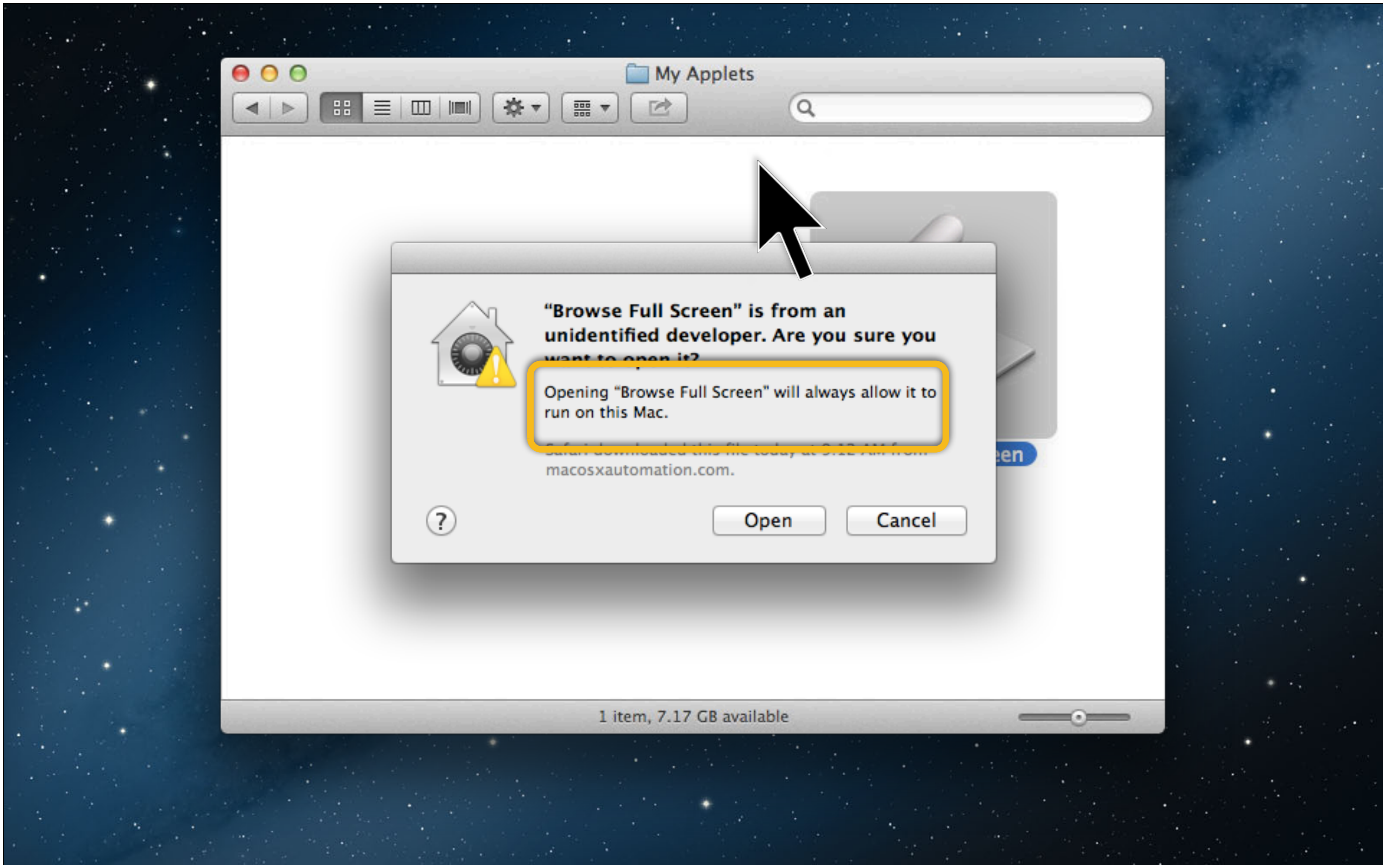
Safari d e today at 9:12 AM from
macosx .


My Applets
Browse Full Screen
1 item, 7.17 GB available


“Browse Full Screen” is from an unidentified developer. Are you sure you want to open it?

Opening “Browse Full Screen” will always allow it to run on this Mac.

Safari downloaded this file today at 9:12 AM from macosxautomation.com.

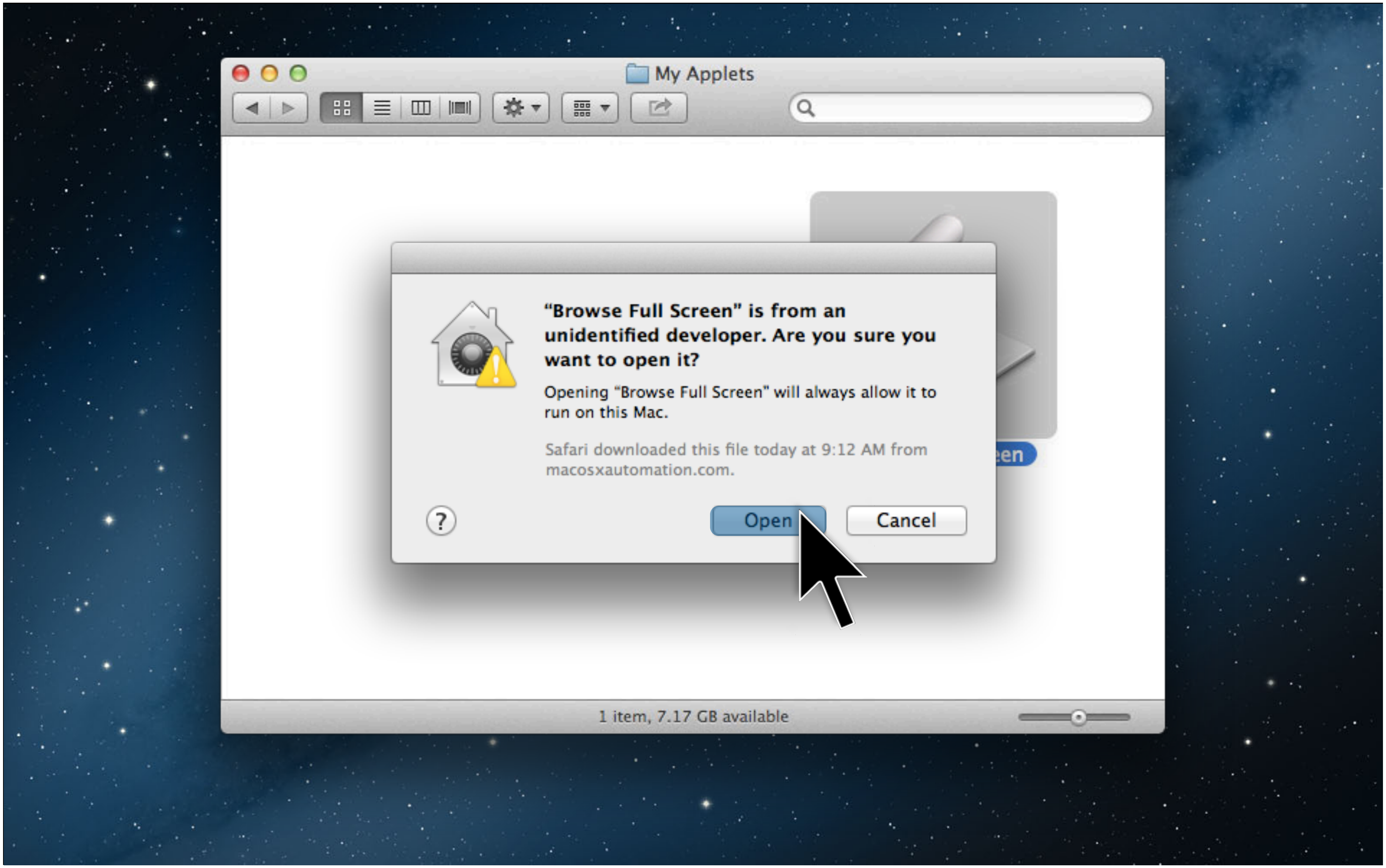


My Applets

“Browse Full Screen” is from an unidentified developer. Are you sure you want to open it?

Opening “Browse Full Screen” will always allow it to run on this Mac.

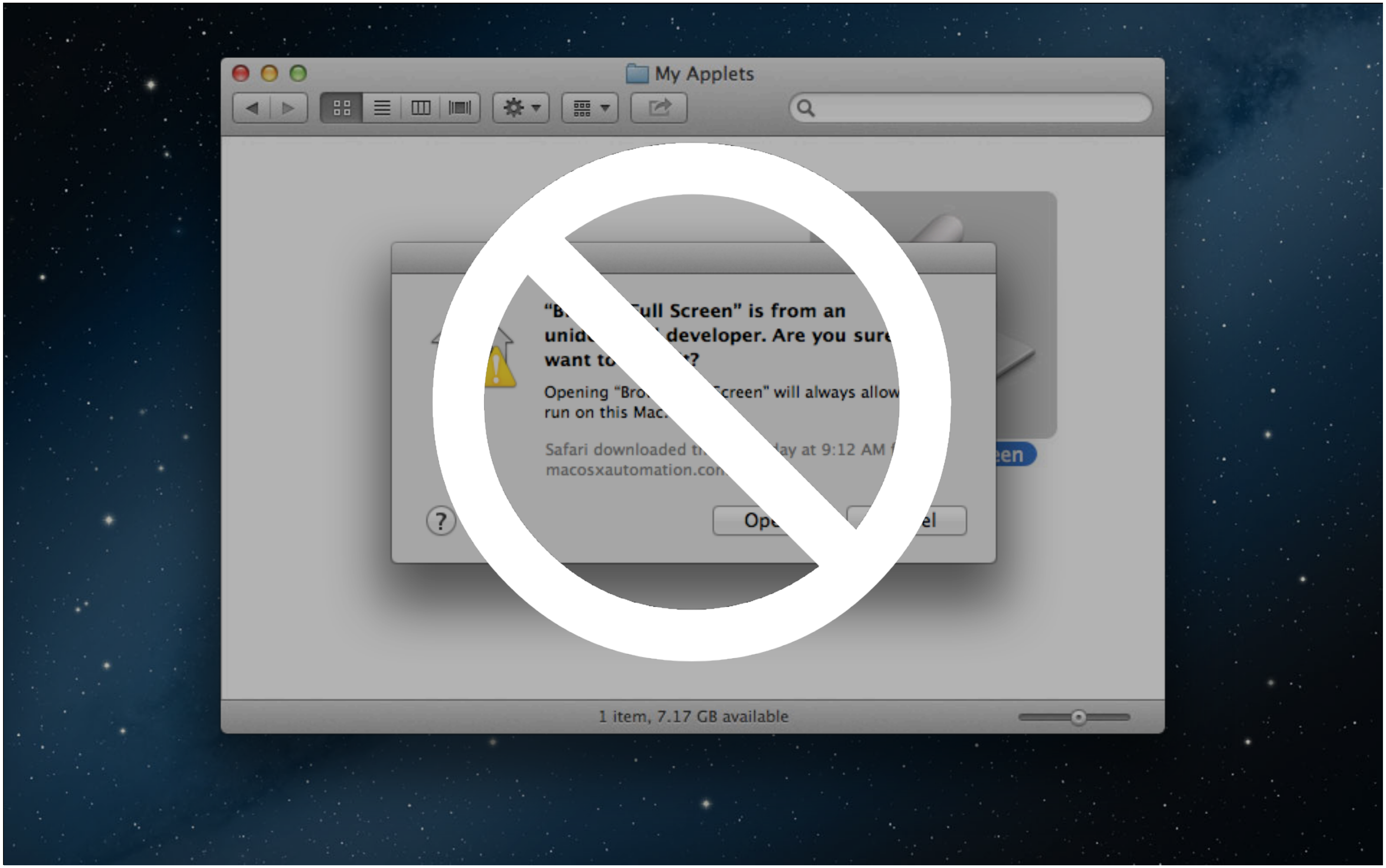
Safari downloaded this file today at 9:12 AM from macosxautomation.com.



"Browse Full Screen" is from an unidentified developer. Are you sure you want to open it?

Opening "Browse Full Screen" will always allow it to run on this Mac.

Safari downloaded this file today at 9:12 AM from macosxautomation.com.



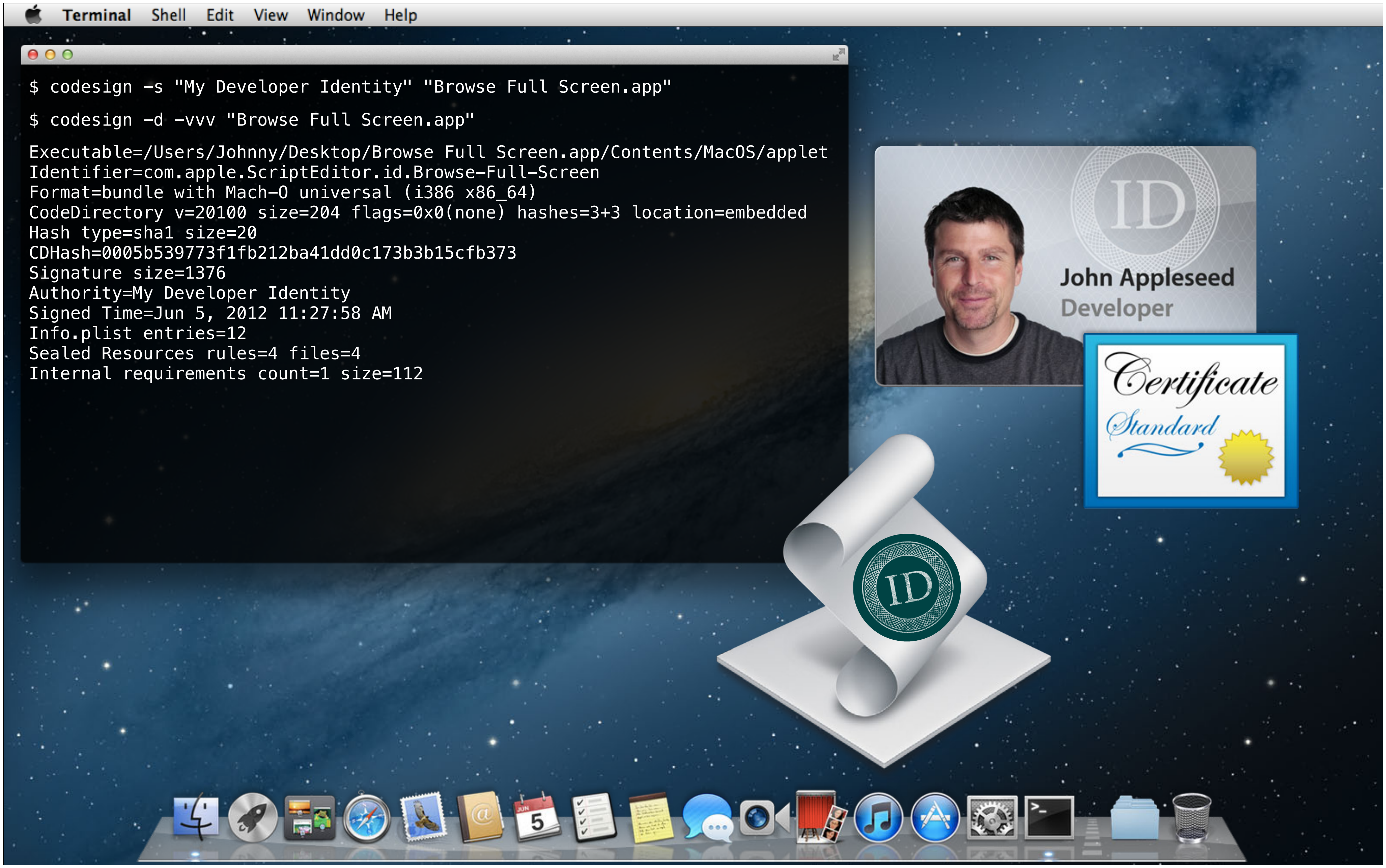
"B… Full Screen" is from an unid… developer. Are you sure want to …t?

Opening "Bro… Screen" will always allow run on this Mac.

Safari downloaded th… day at 9:12 AM f… macosxautomation.com

Terminal
Shell
Edit
View
Window
Help
ID
John Appleseed
Developer

Terminal
Shell
Edit
View
Window
Help
ID
John Appleseed
Developer
Certificate
Standard



```
$ codesign -s "My Developer Identity" "Browse Full Screen.app"

$ codesign -d -vvv "Browse Full Screen.app"

Executable=/Users/Johnny/Desktop/Browse Full Screen.app/Contents/MacOS/applet
Identifier=com.apple.ScriptEditor.id.Browse-Full-Screen
Format=bundle with Mach-O universal (i386 x86_64)
CodeDirectory v=20100 size=204 flags=0x0(none) hashes=3+3 location=embedded
Hash type=sha1 size=20
CDHash=0005b539773f1fb212ba41dd0c173b3b15cfb373
Signature size=1376
Authority=My Developer Identity
Signed Time=Jun 5, 2012 11:27:58 AM
Info.plist entries=12
Sealed Resources rules=4 files=4
Internal requirements count=1 size=112
```

Demo

Signing Automation Applets

Chris Nebel
Senior Engineer Automation Technologies

Retrieving Scripts from Web Links

AppleScript URL Protocol

AppleScript URL Protocol

Easy-to-share AppleScript sample code

AppleScript URL Protocol

Easy-to-share AppleScript sample code

- AppleScript sample code placed in webpage links

AppleScript URL Protocol

Easy-to-share AppleScript sample code

- AppleScript sample code placed in webpage links
- URLs begin with

```
applescript://
```

AppleScript URL Protocol

Easy-to-share AppleScript sample code

- AppleScript sample code placed in webpage links
- URLs begin with

  `applescript://`
- Script code encoded as a URL query parameter

AppleScript URL Protocol

Easy-to-share AppleScript sample code

- AppleScript sample code placed in webpage links
- URLs begin with

```
applescript://
```

- Script code encoded as a URL query parameter
- Example (“Hello World”)

```
applescript://com.apple.scripteditor?action=new&script=display%20dialog
%20%22Hello%20World%22
```

AppleScript URL Protocol

Easy-to-share AppleScript sample code

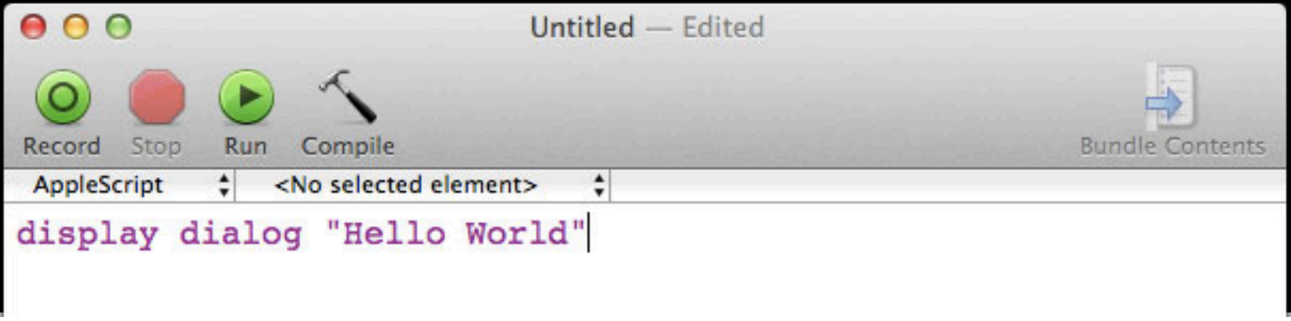
- AppleScript sample code placed in webpage links
- URLs begin with

```
applescript://
```

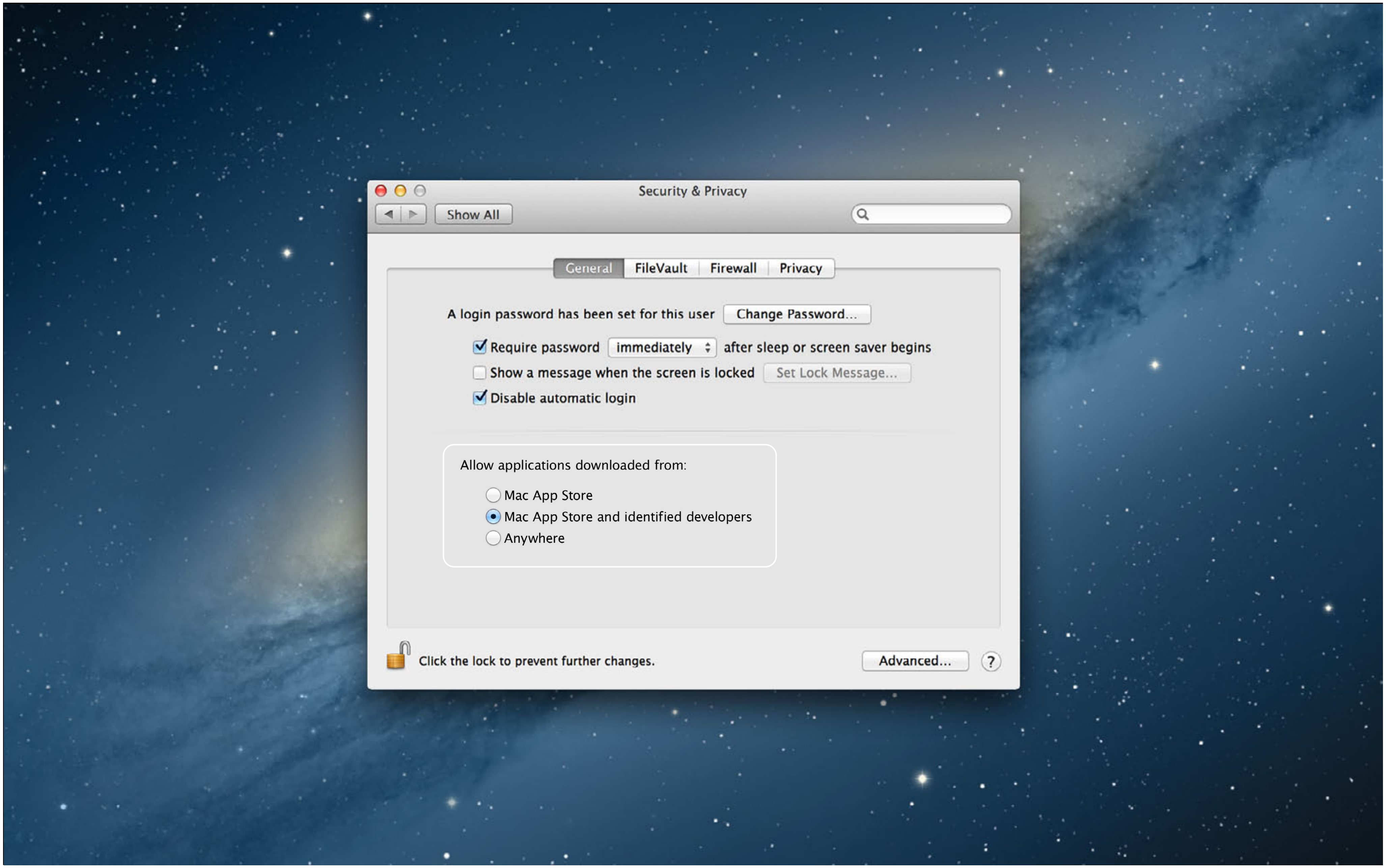
- Script code encoded as a URL query parameter
- Example (“Hello World”)

```
applescript://com.apple.scripteditor?action=new&script=display%20dialog
%20%22Hello%20World%22
```

- Clicked link opens code in AppleScript Editor



```
display dialog "Hello World"
```



A login password has been set for this user

[✓] Require password immediately after sleep or screen saver begins

[ ] Show a message when the screen is locked

[✓] Disable automatic login

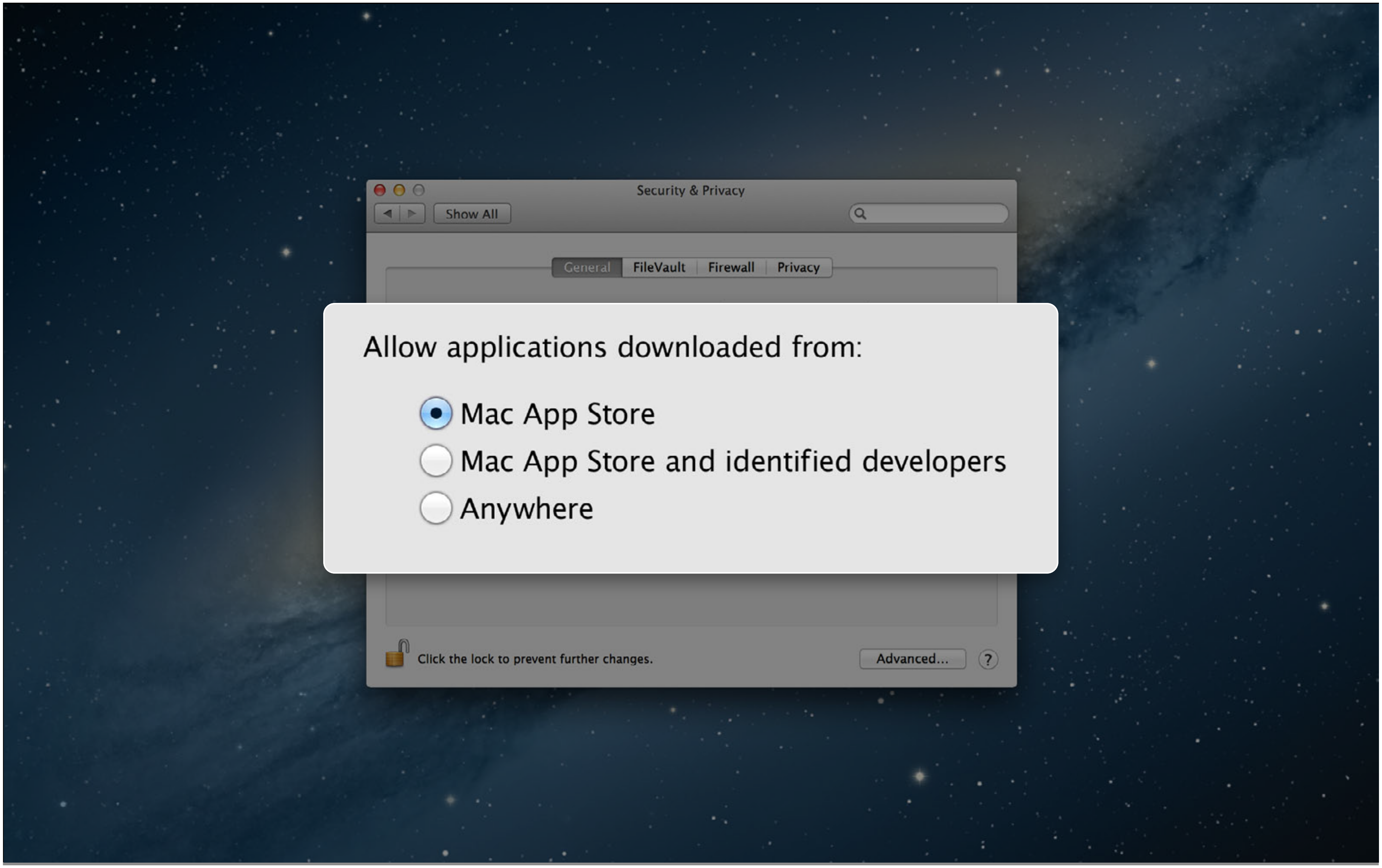
Allow applications downloaded from:

( ) Mac App Store

(●) Mac App Store and identified developers

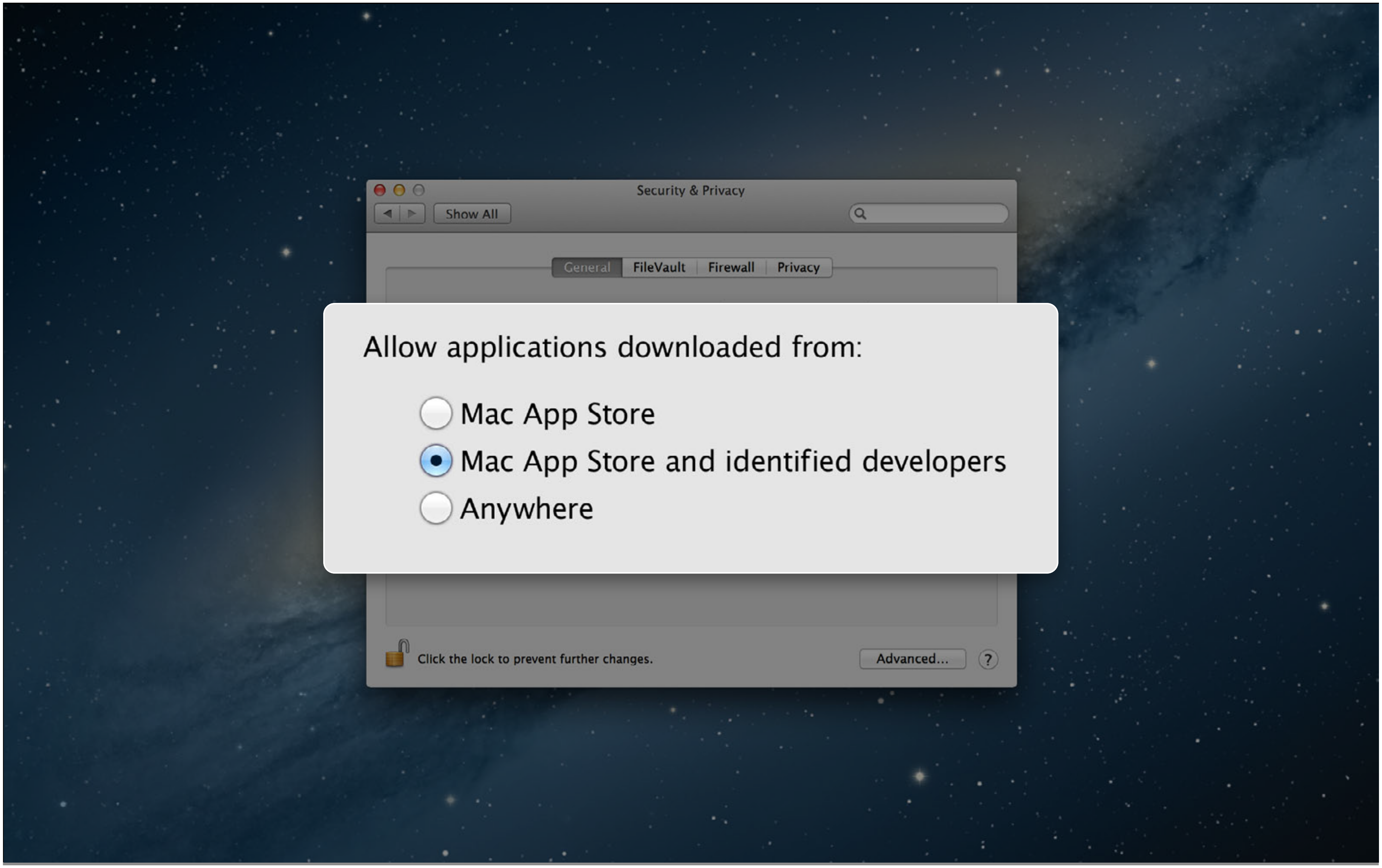
( ) Anywhere

Click the lock to prevent further changes.



Allow applications downloaded from:

[✓ Mac App Store]
Mac App Store and identified developers
Anywhere


Security & Privacy

Show All

General | FileVault | Firewall | Privacy

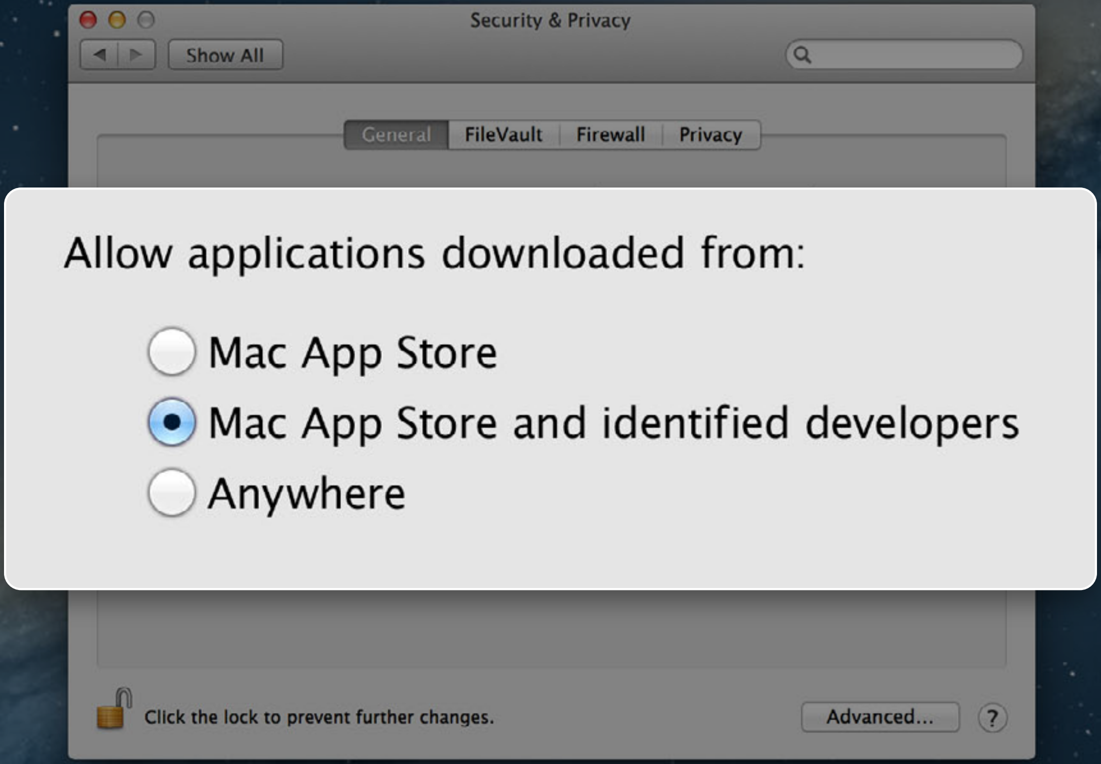
Allow applications downloaded from:

- ( ) Mac App Store
- [✓ Mac App Store and identified developers]
- ( ) Anywhere

Click the lock to prevent further changes.

Advanced... ?

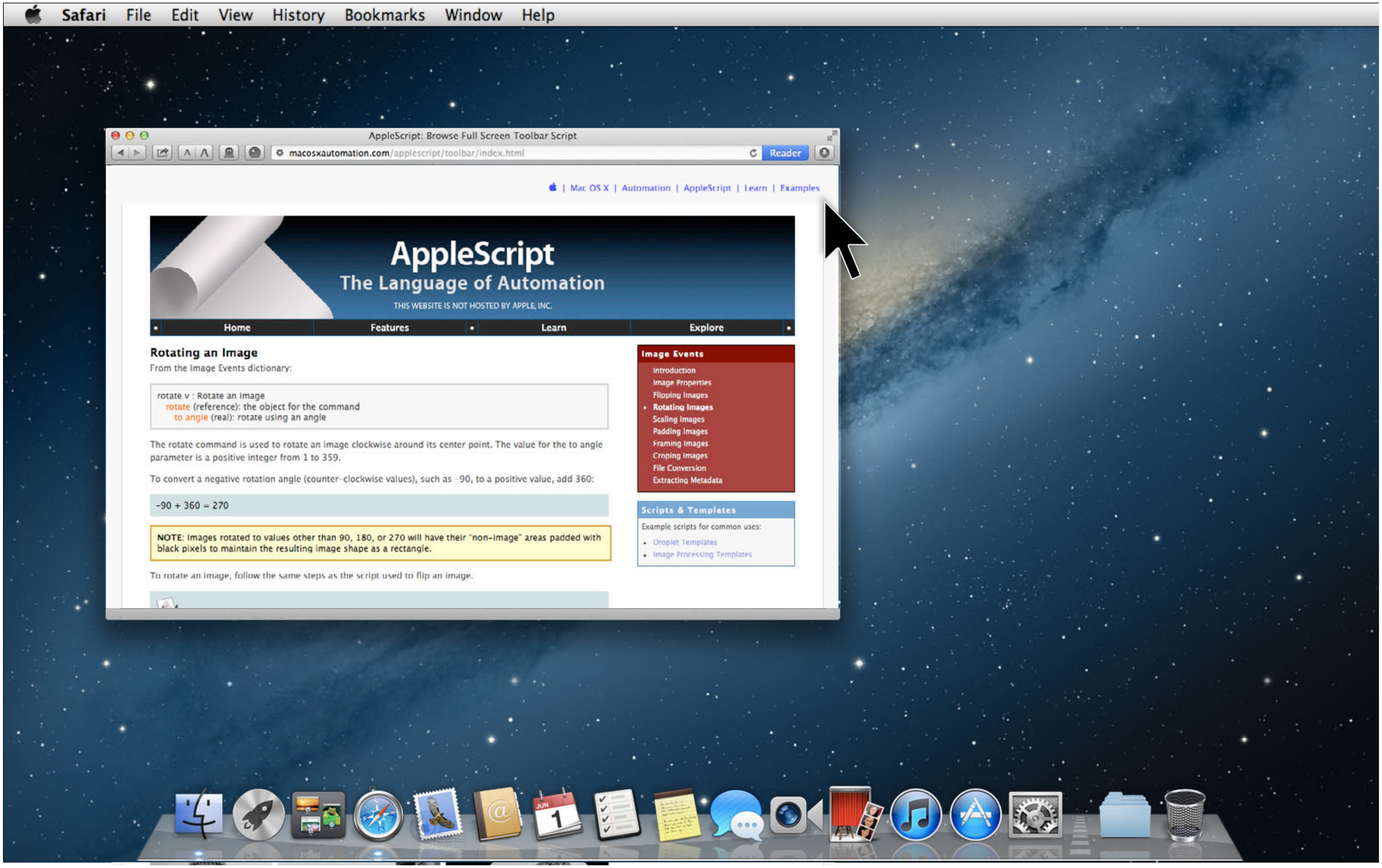
Default Setting



Allow applications downloaded from:

- ○ Mac App Store
- [✓ ● Mac App Store and identified developers]
- ○ Anywhere

Click the lock to prevent further changes.



Safari File Edit View History Bookmarks Window Help
AppleScript: Browse Full Screen Toolbar Script
macosxautomation.com/applescript/toolbar/index.html
Reader
| Mac OS X | Automation | AppleScript | Learn | Examples
AppleScript
The Language of Automation
THIS WEBSITE IS NOT HOSTED BY APPLE, INC.
Home
Features
Learn
Explore

Rotating an Image

From the Image Events dictionary:

```
rotate v : Rotate an image
  rotate (reference): the object for the command
    to angle (real): rotate using an angle
```

The rotate command is used to rotate an image clockwise around its center point. The value for the to angle parameter is a positive integer from 1 to 359.

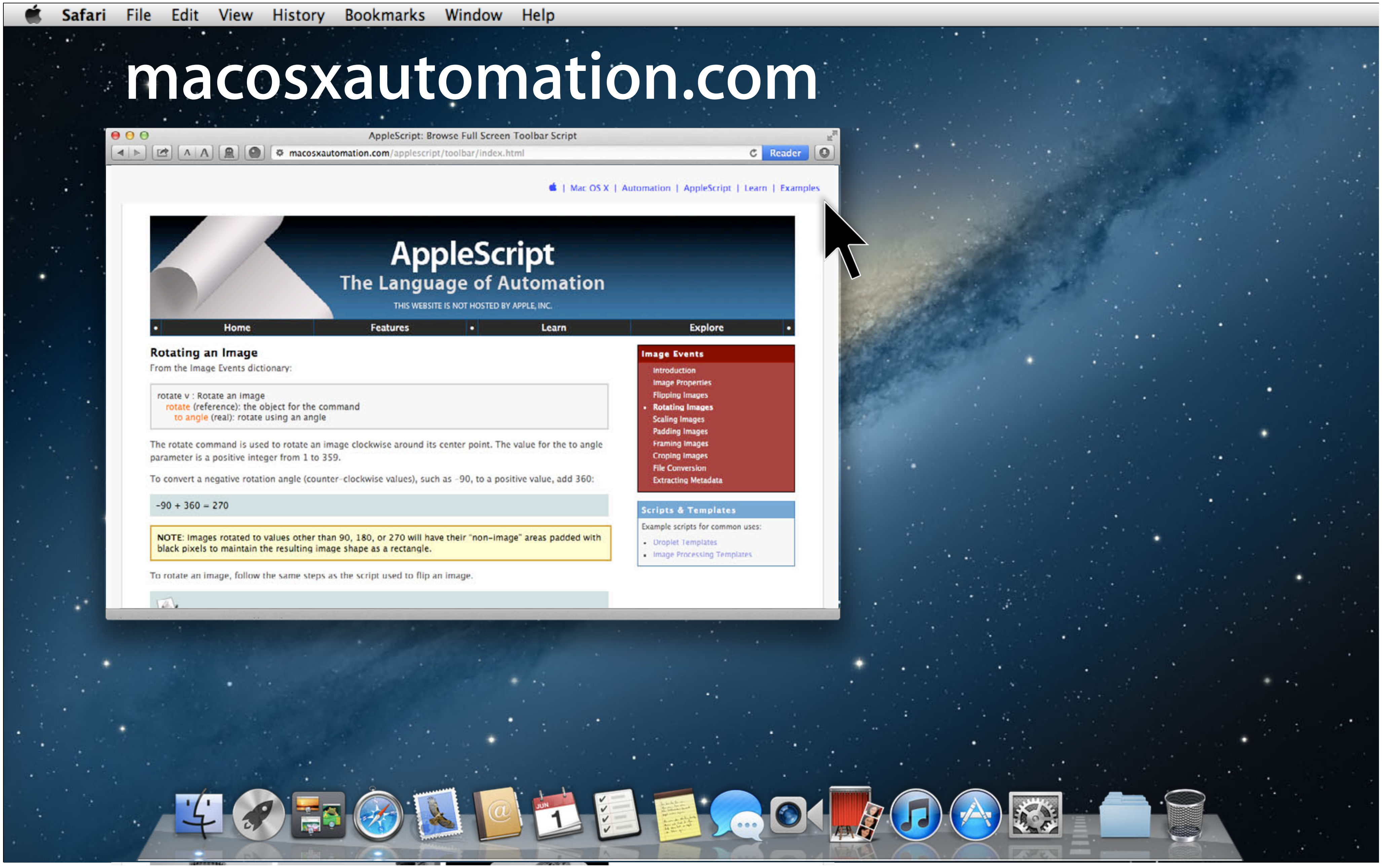
To convert a negative rotation angle (counter-clockwise values), such as -90, to a positive value, add 360:

```
-90 + 360 = 270
```

NOTE: Images rotated to values other than 90, 180, or 270 will have their "non-image" areas padded with black pixels to maintain the resulting image shape as a rectangle.

To rotate an image, follow the same steps as the script used to flip an image.

macosxautomation.com



AppleScript: Browse Full Screen Toolbar Script

macosxautomation.com/applescript/toolbar/index.html

| Mac OS X | Automation | AppleScript | Learn | Examples

AppleScript
The Language of Automation
THIS WEBSITE IS NOT HOSTED BY APPLE, INC.

Rotating an Image

From the Image Events dictionary:

```
rotate v : Rotate an image
  rotate (reference): the object for the command
    to angle (real): rotate using an angle
```

The rotate command is used to rotate an image clockwise around its center point. The value for the to angle parameter is a positive integer from 1 to 359.

To convert a negative rotation angle (counter-clockwise values), such as -90, to a positive value, add 360:

```
-90 + 360 = 270
```

NOTE: Images rotated to values other than 90, 180, or 270 will have their "non-image" areas padded with black pixels to maintain the resulting image shape as a rectangle.

To rotate an image, follow the same steps as the script used to flip an image.

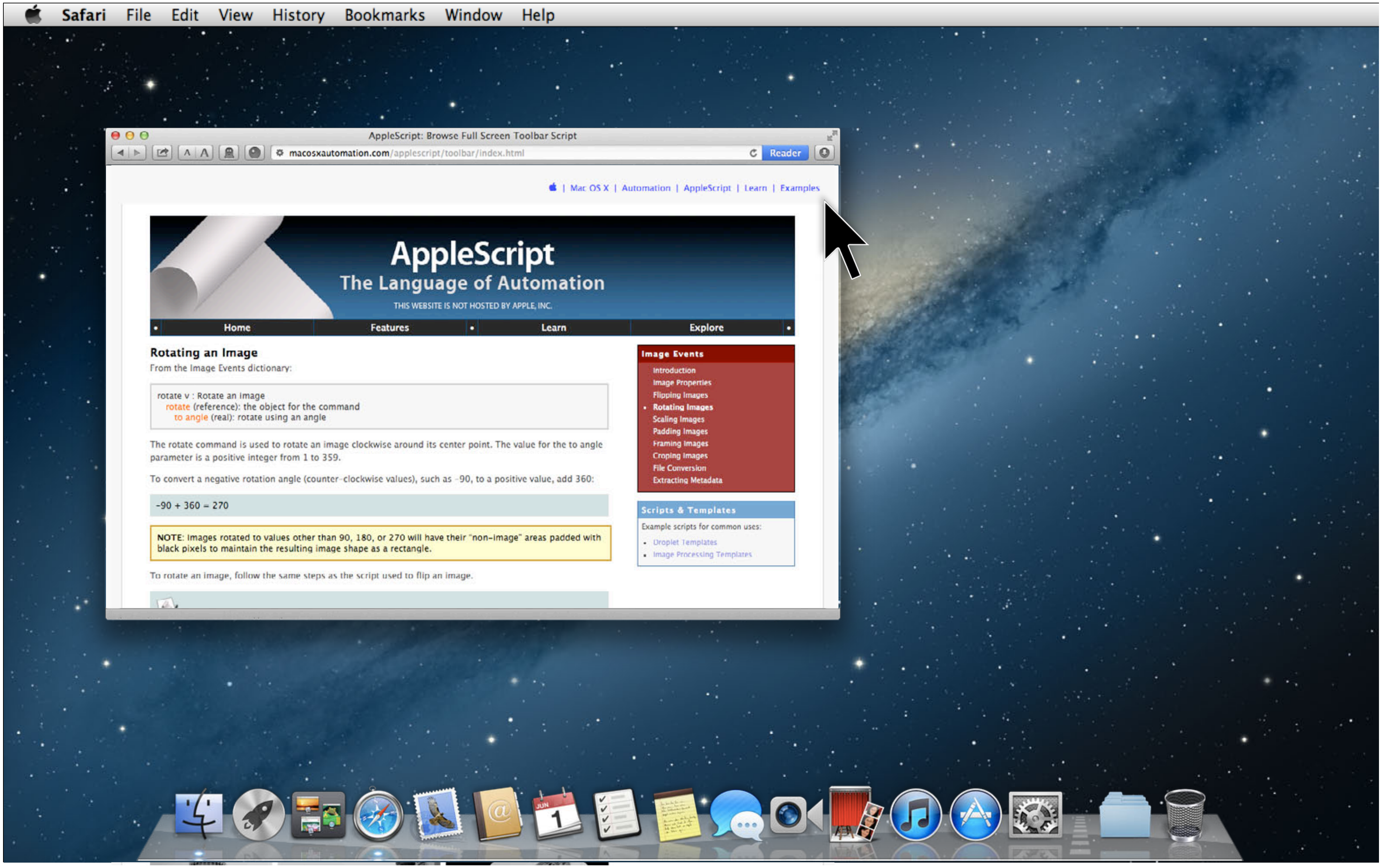
Image Events

- Introduction
- Image Properties
- Flipping Images
- Rotating Images
- Scaling Images
- Padding Images
- Framing Images
- Croping Images
- File Conversion
- Extracting Metadata

Scripts & Templates

Example scripts for common uses:

- Droplet Templates
- Image Processing Templates


Safari File Edit View History Bookmarks Window Help
AppleScript: Browse Full Screen Toolbar Script
macosxautomation.com/applescript/toolbar/index.html
Reader
| Mac OS X | Automation | AppleScript | Learn | Examples
AppleScript
The Language of Automation
THIS WEBSITE IS NOT HOSTED BY APPLE, INC.
Home
Features
Learn
Explore
JUN 1

Rotating an Image

From the Image Events dictionary:

```
rotate v : Rotate an image
  rotate (reference): the object for the command
    to angle (real): rotate using an angle
```

The rotate command is used to rotate an image clockwise around its center point. The value for the to angle parameter is a positive integer from 1 to 359.

To convert a negative rotation angle (counter-clockwise values), such as -90, to a positive value, add 360:

```
-90 + 360 = 270
```

NOTE: Images rotated to values other than 90, 180, or 270 will have their "non-image" areas padded with black pixels to maintain the resulting image shape as a rectangle.

To rotate an image, follow the same steps as the script used to flip an image.

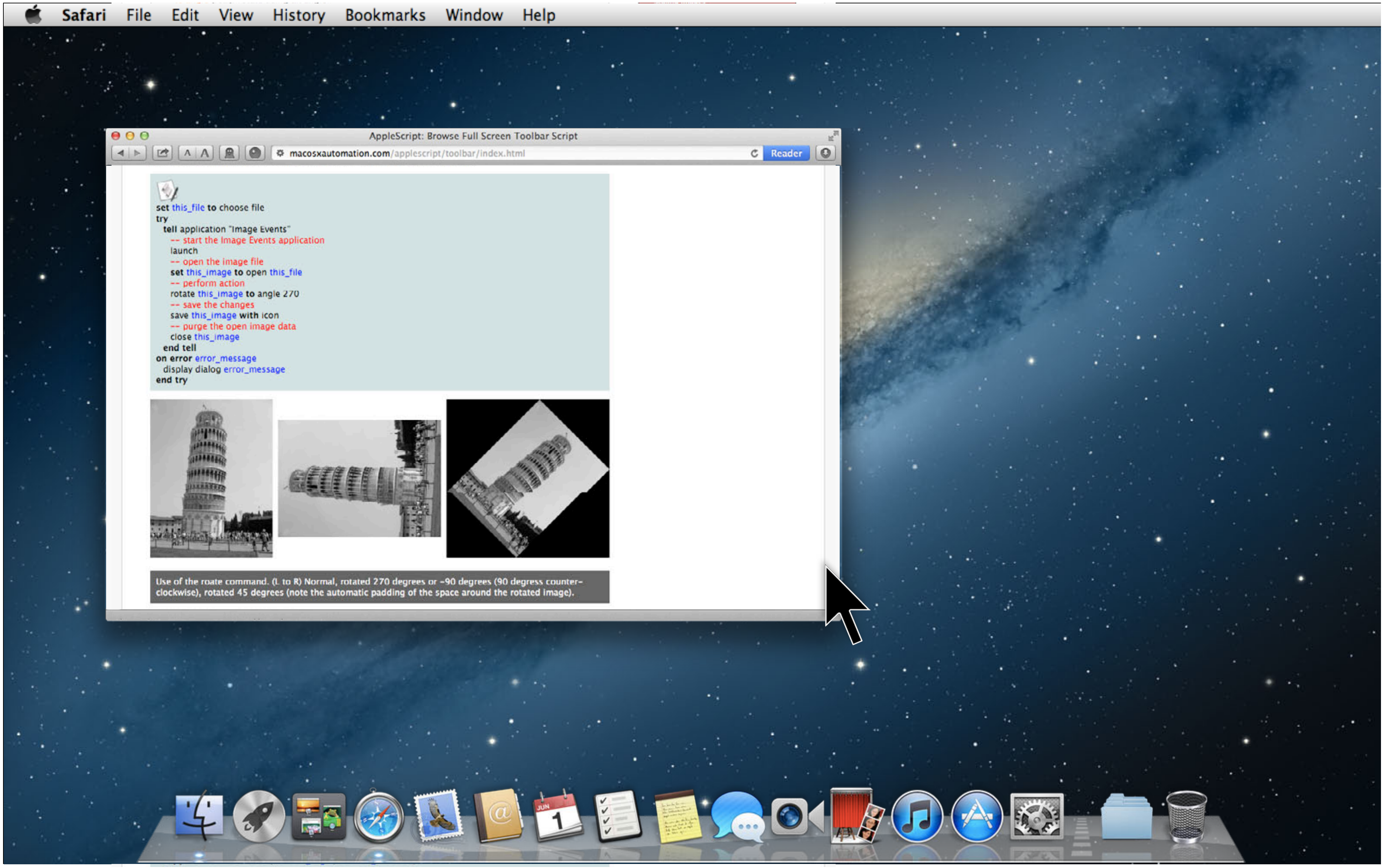
Image Events

- Introduction
- Image Properties
- Flipping Images
- Rotating Images
- Scaling Images
- Padding Images
- Framing Images
- Croping Images
- File Conversion
- Extracting Metadata

Scripts & Templates

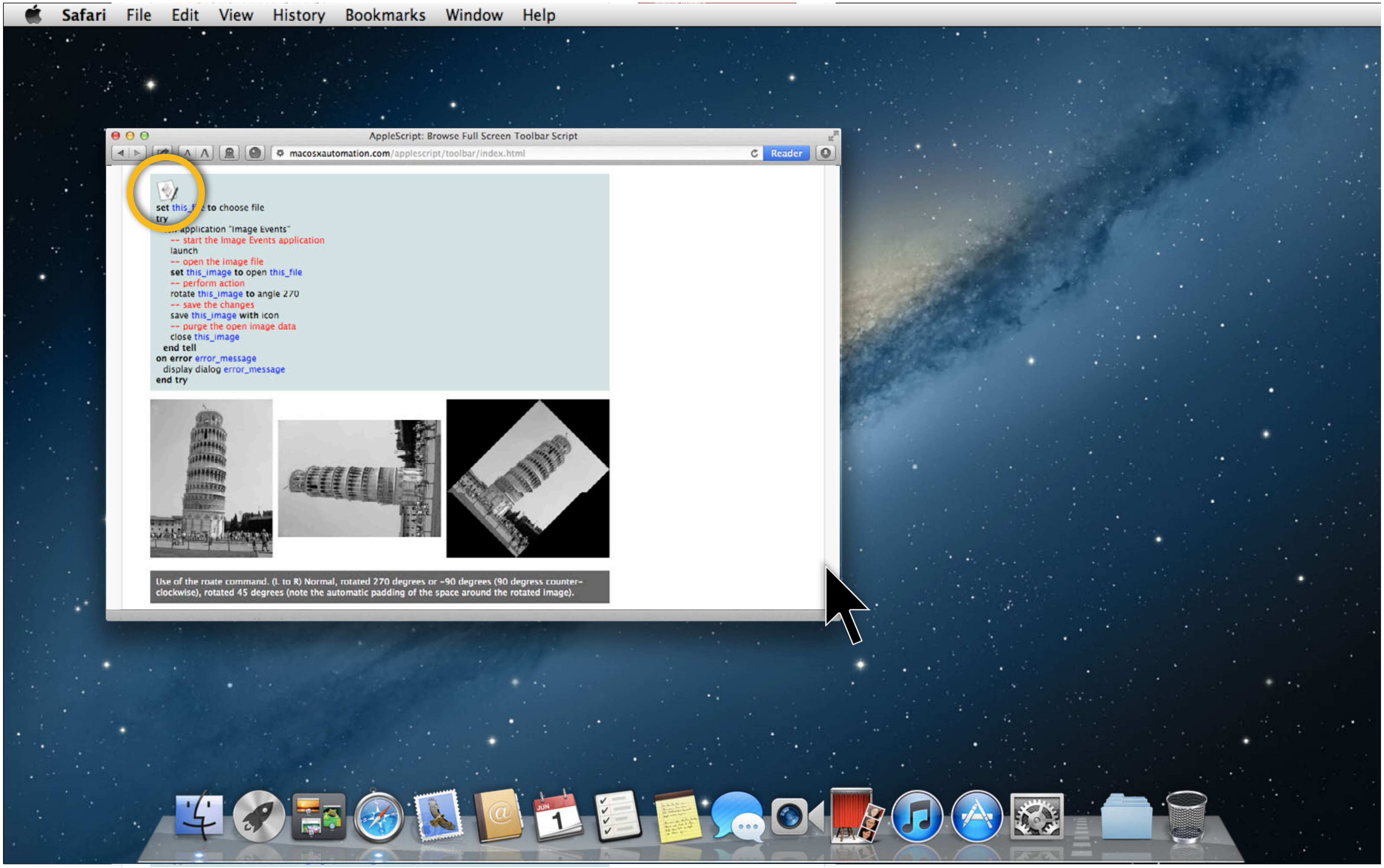
Example scripts for common uses:

- Droplet Templates
- Image Processing Templates


Safari
File
Edit
View
History
Bookmarks
Window
Help
AppleScript: Browse Full Screen Toolbar Script
macosxautomation.com/applescript/toolbar/index.html
Reader

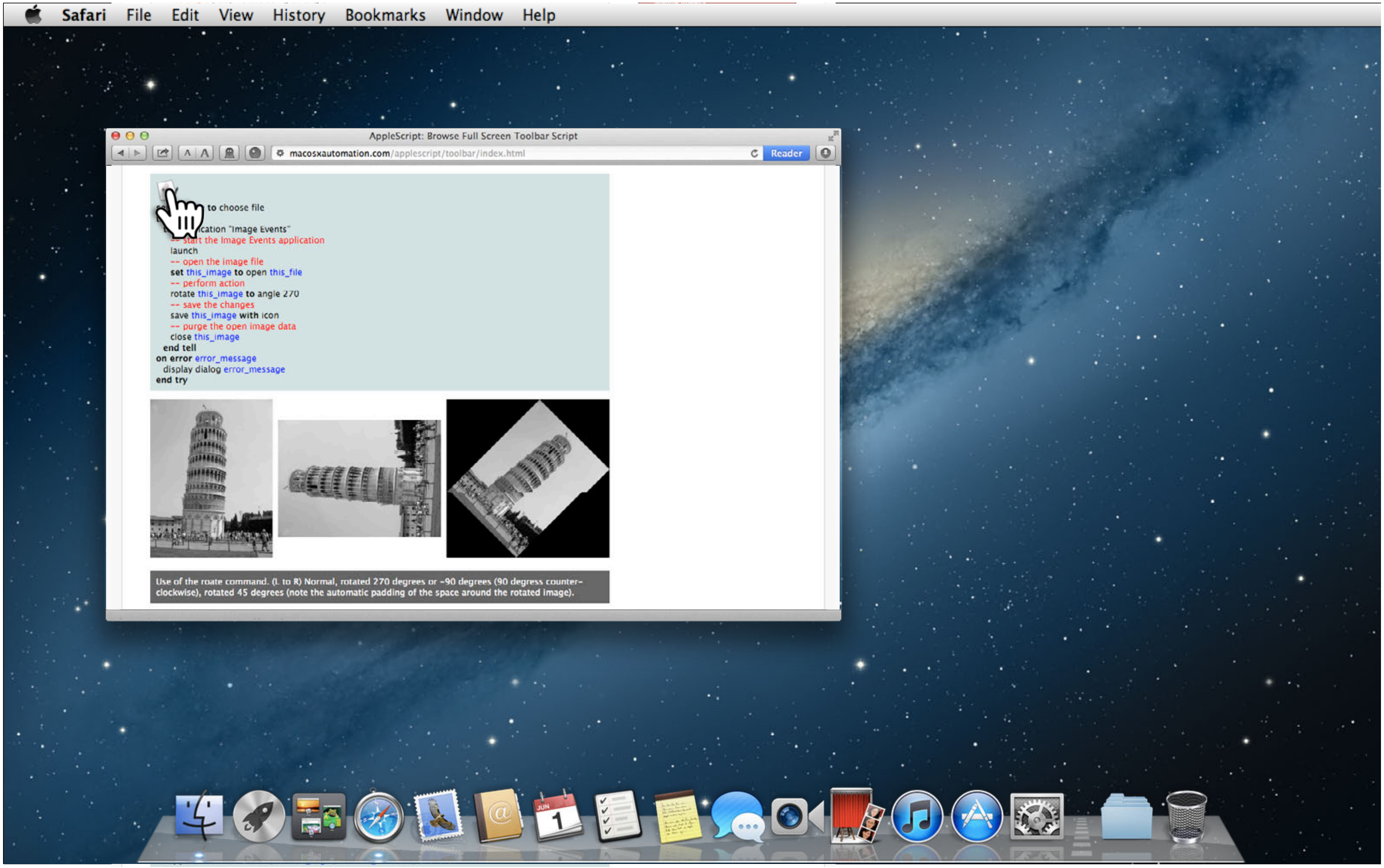
```
set this_file to choose file
try
  tell application "Image Events"
    -- start the Image Events application
    launch
    -- open the image file
    set this_image to open this_file
    -- perform action
    rotate this_image to angle 270
    -- save the changes
    save this_image with icon
    -- purge the open image data
    close this_image
  end tell
on error error_message
  display dialog error_message
end try
```

Use of the roate command. (L to R) Normal, rotated 270 degrees or -90 degrees (90 degress counter-clockwise), rotated 45 degrees (note the automatic padding of the space around the rotated image).


Safari
File
Edit
View
History
Bookmarks
Window
Help
AppleScript: Browse Full Screen Toolbar Script
macosxautomation.com/applescript/toolbar/index.html
Reader

```
set this_file to choose file
try
  tell application "Image Events"
    -- start the Image Events application
    launch
    -- open the image file
    set this_image to open this_file
    -- perform action
    rotate this_image to angle 270
    -- save the changes
    save this_image with icon
    -- purge the open image data
    close this_image
  end tell
on error error_message
  display dialog error_message
end try
```

Use of the roate command. (L to R) Normal, rotated 270 degrees or -90 degrees (90 degress counter-clockwise), rotated 45 degrees (note the automatic padding of the space around the rotated image).

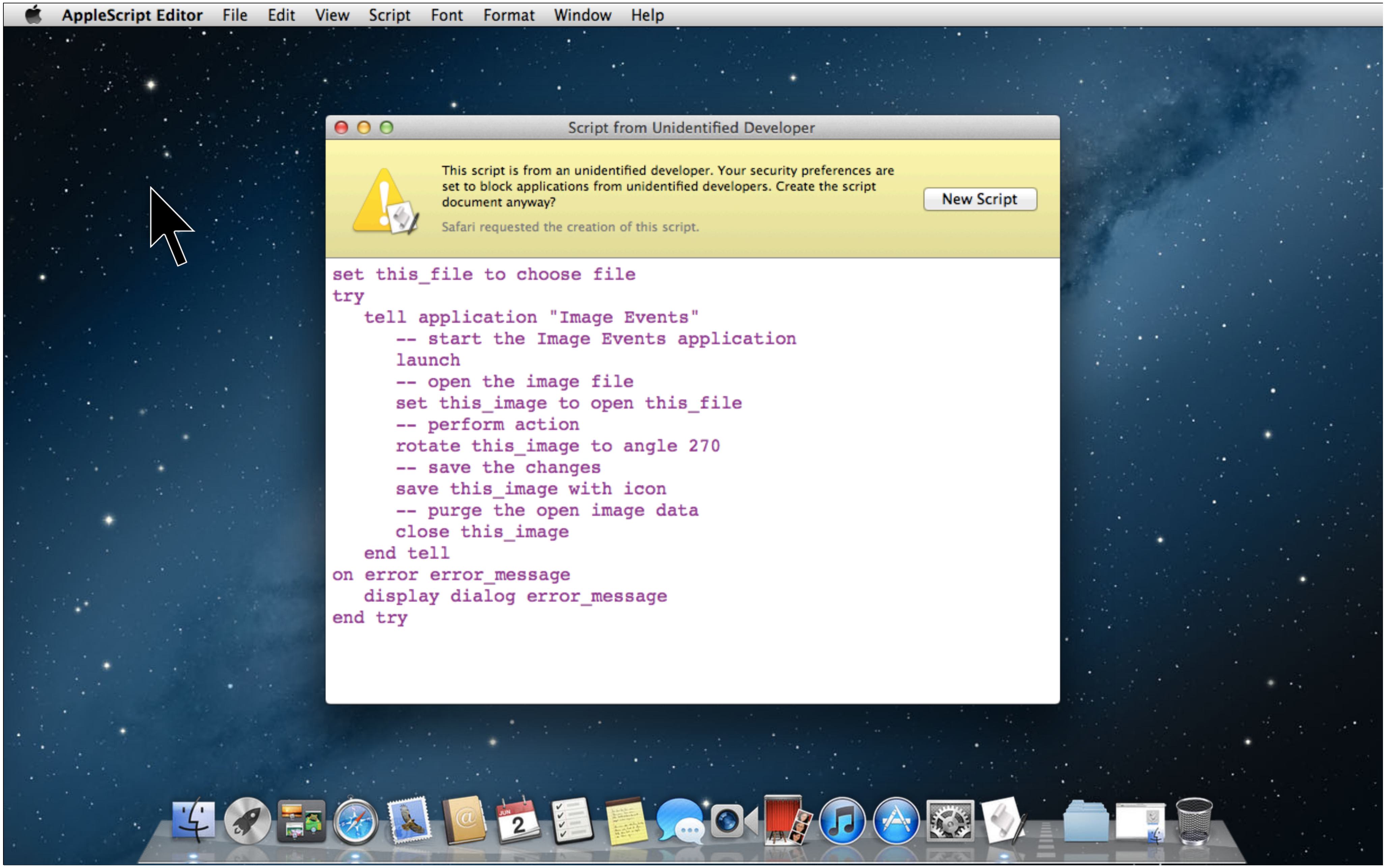

Safari
File
Edit
View
History
Bookmarks
Window
Help
AppleScript: Browse Full Screen Toolbar Script
macosxautomation.com/applescript/toolbar/index.html
Reader

```
 to choose file

    ication "Image Events"
    -- start the Image Events application
    launch
    -- open the image file
    set this_image to open this_file
    -- perform action
    rotate this_image to angle 270
    -- save the changes
    save this_image with icon
    -- purge the open image data
    close this_image
  end tell
on error error_message
  display dialog error_message
end try
```

Use of the roate command. (L to R) Normal, rotated 270 degrees or -90 degrees (90 degrees counter-clockwise), rotated 45 degrees (note the automatic padding of the space around the rotated image).

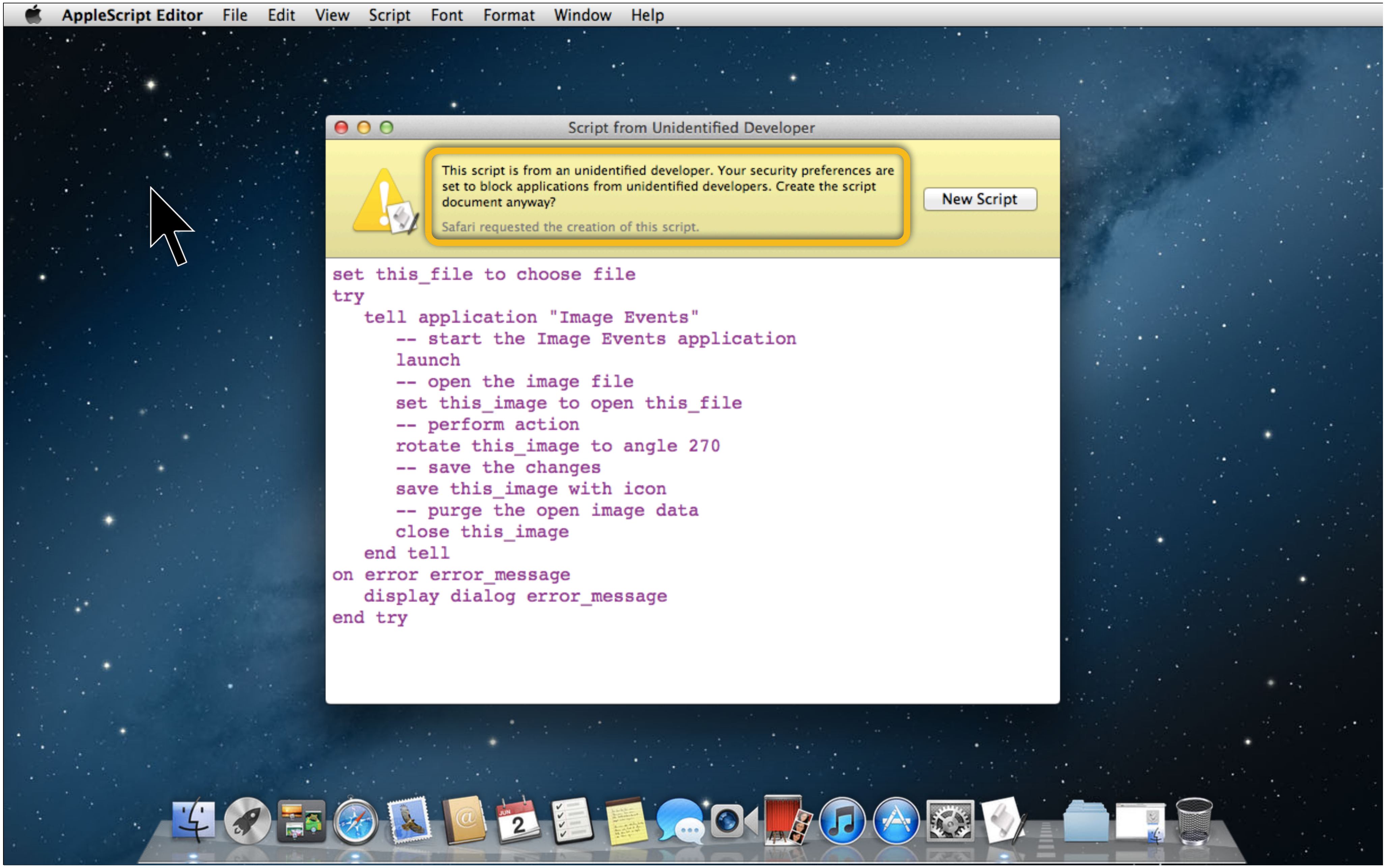
AppleScript Editor
File
Edit
View
Script
Font
Format
Window
Help



This script is from an unidentified developer. Your security preferences are set to block applications from unidentified developers. Create the script document anyway?

Safari requested the creation of this script.

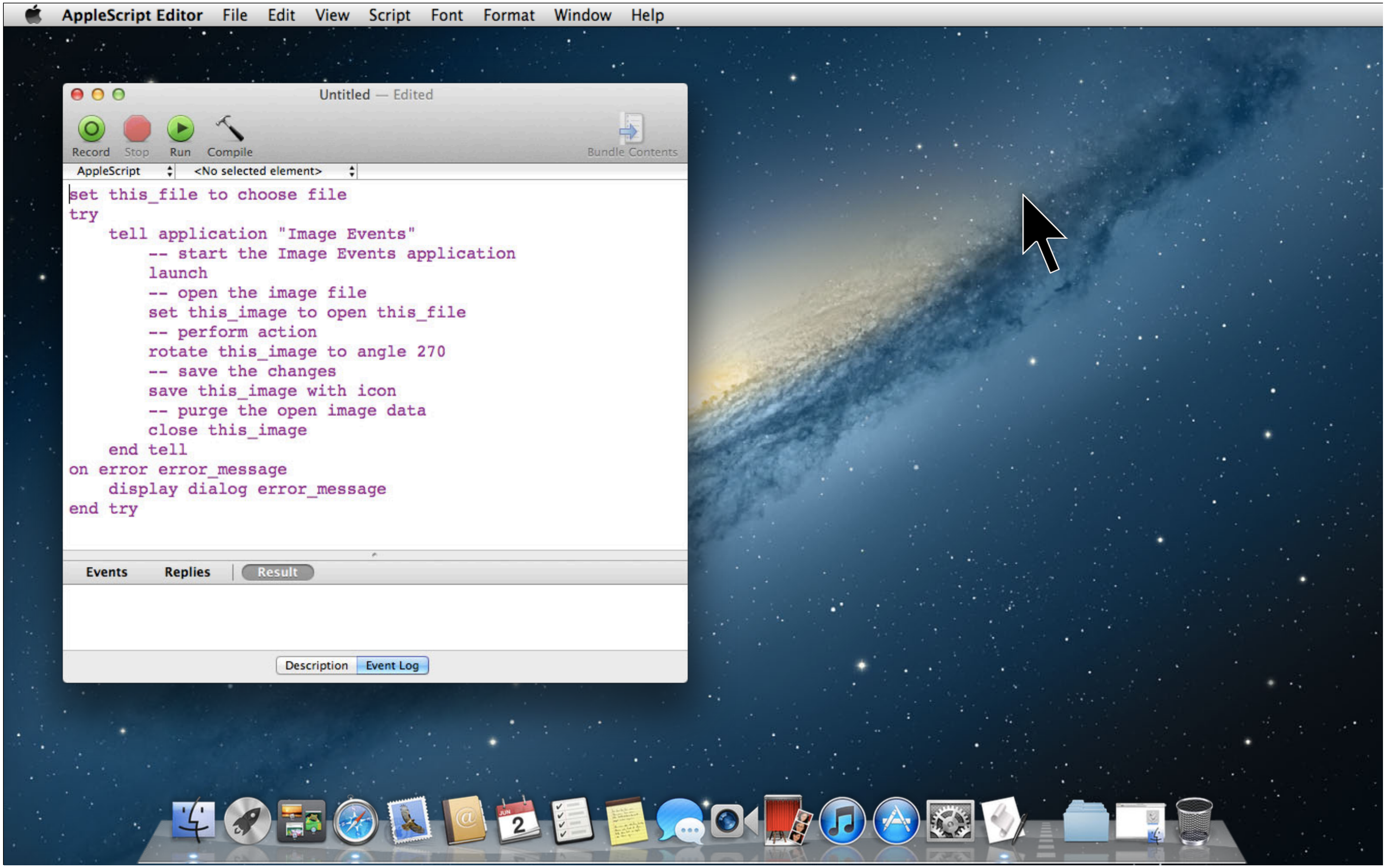
```
set this_file to choose file
try
   tell application "Image Events"
      -- start the Image Events application
      launch
      -- open the image file
      set this_image to open this_file
      -- perform action
      rotate this_image to angle 270
      -- save the changes
      save this_image with icon
      -- purge the open image data
      close this_image
   end tell
on error error_message
   display dialog error_message
end try
```


AppleScript Editor File Edit View Script Font Format Window Help
Script from Unidentified Developer
New Script

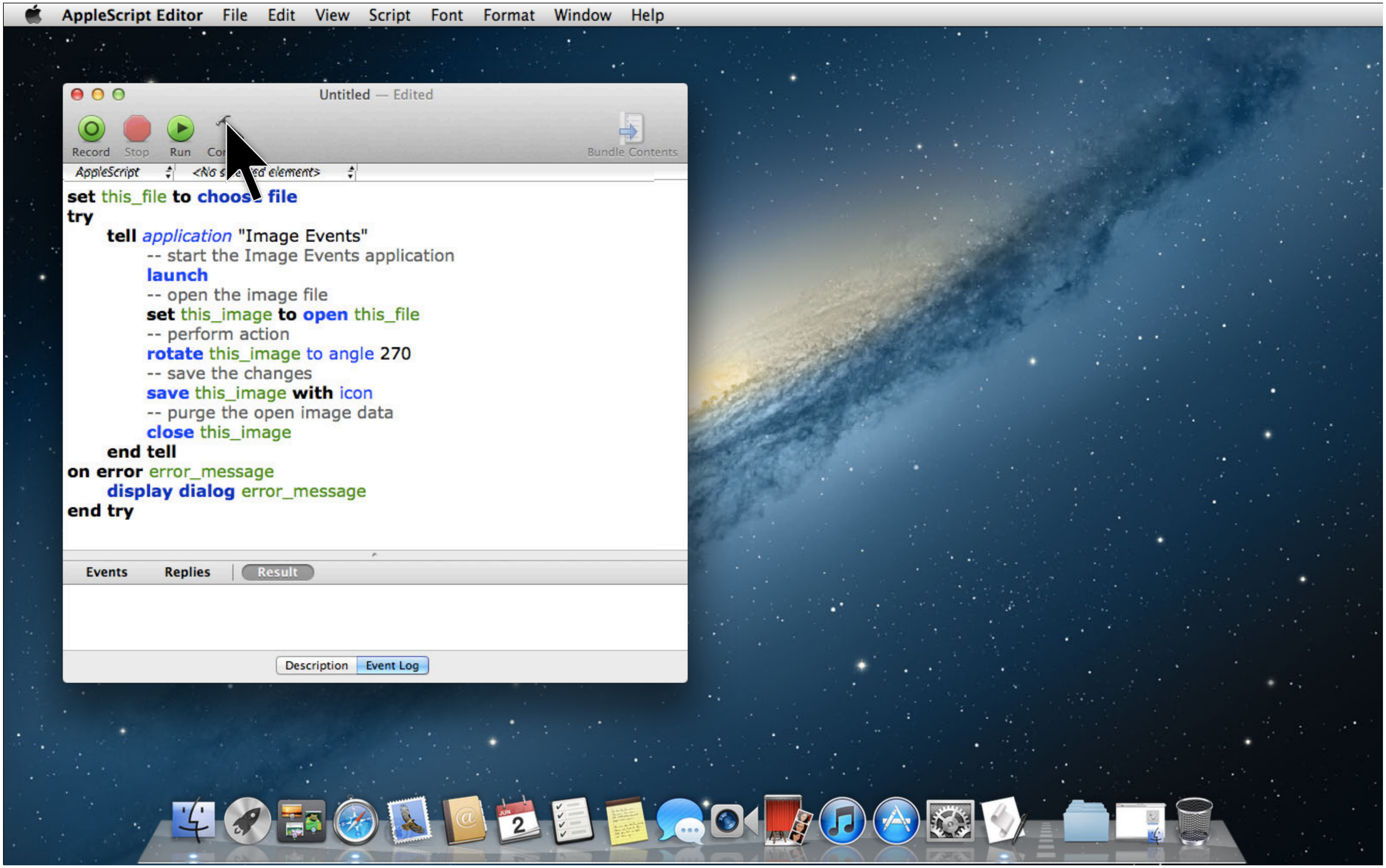
This script is from an unidentified developer. Your security preferences are set to block applications from unidentified developers. Create the script document anyway?

Safari requested the creation of this script.

```
set this_file to choose file
try
   tell application "Image Events"
      -- start the Image Events application
      launch
      -- open the image file
      set this_image to open this_file
      -- perform action
      rotate this_image to angle 270
      -- save the changes
      save this_image with icon
      -- purge the open image data
      close this_image
   end tell
on error error_message
   display dialog error_message
end try
```



```
set this_file to choose file
try
    tell application "Image Events"
        -- start the Image Events application
        launch
        -- open the image file
        set this_image to open this_file
        -- perform action
        rotate this_image to angle 270
        -- save the changes
        save this_image with icon
        -- purge the open image data
        close this_image
    end tell
on error error_message
    display dialog error_message
end try
```



```
set this_file to choos  file
try
	tell application "Image Events"
		-- start the Image Events application
		launch
		-- open the image file
		set this_image to open this_file
		-- perform action
		rotate this_image to angle 270
		-- save the changes
		save this_image with icon
		-- purge the open image data
		close this_image
	end tell
on error error_message
	display dialog error_message
end try
```

Retrieving Scripts from Web Links

AppleScript URL Protocol

Distributing Scripts
Working with Gatekeeper

- Gatekeeper may restrict downloaded applets
    - Sign applets with an Apple developer signature
- Gatekeeper may check `applescript:` URLs

App Sandbox

App-to-App Automation

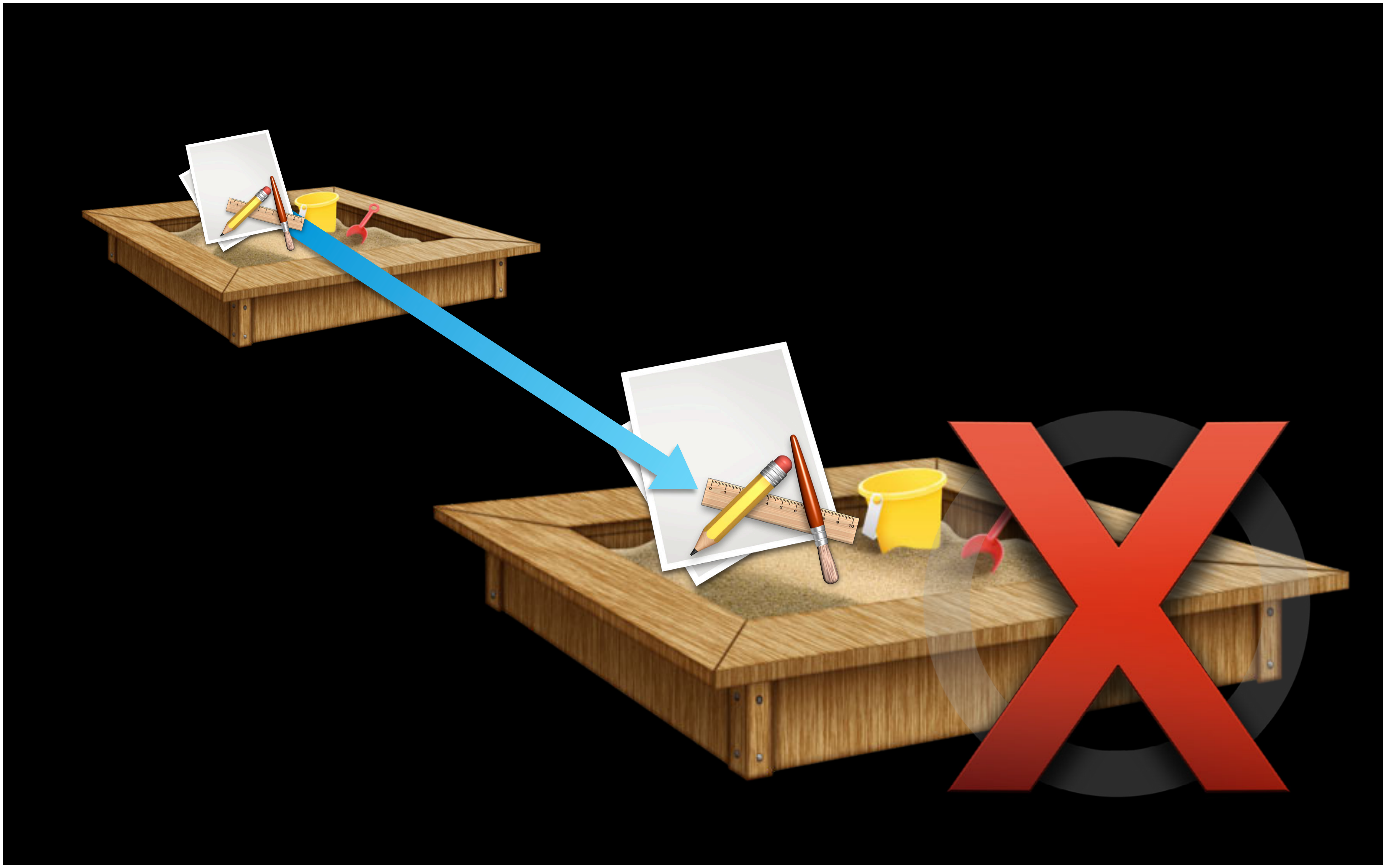
Using Apple events with your application

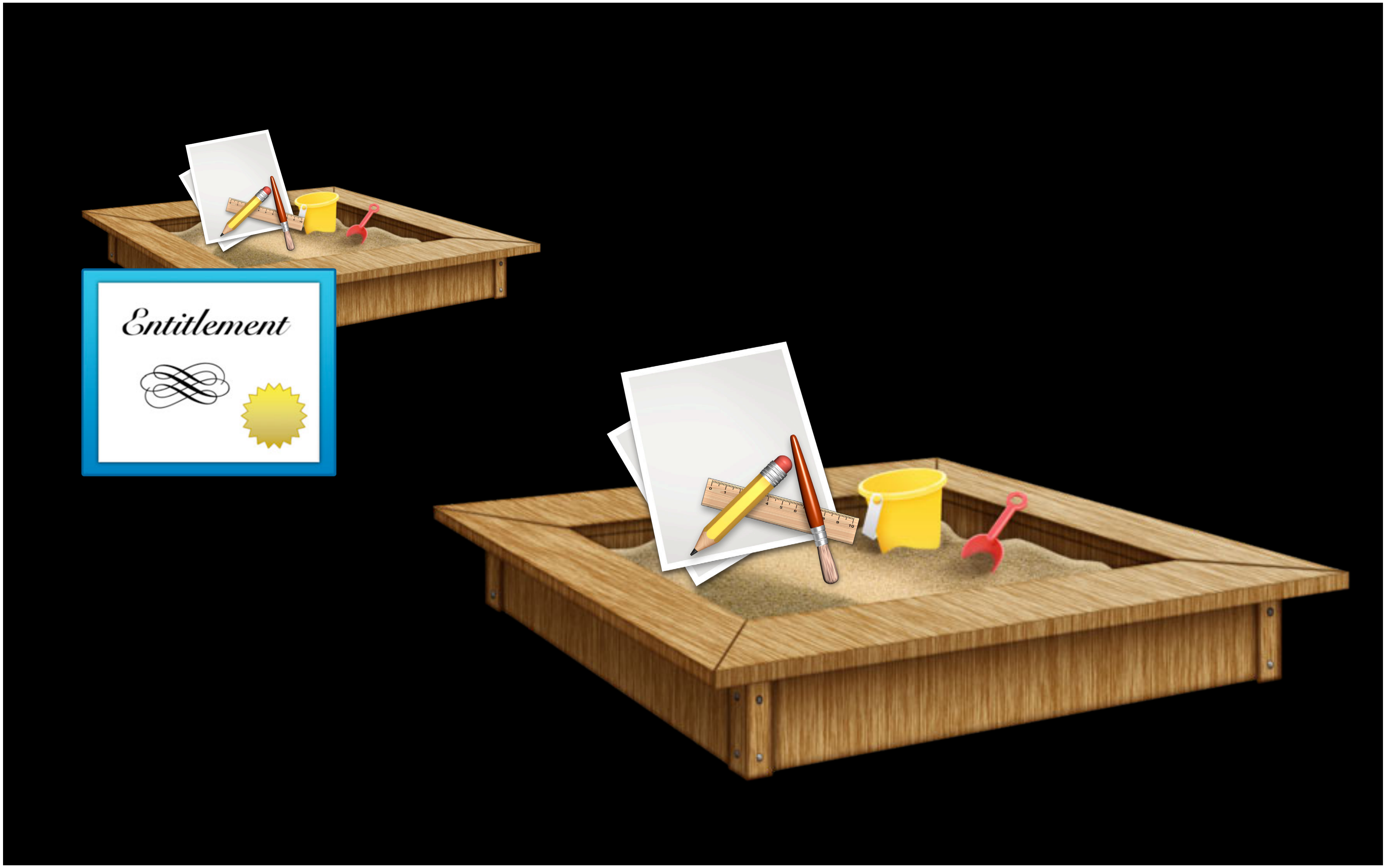
Apple events

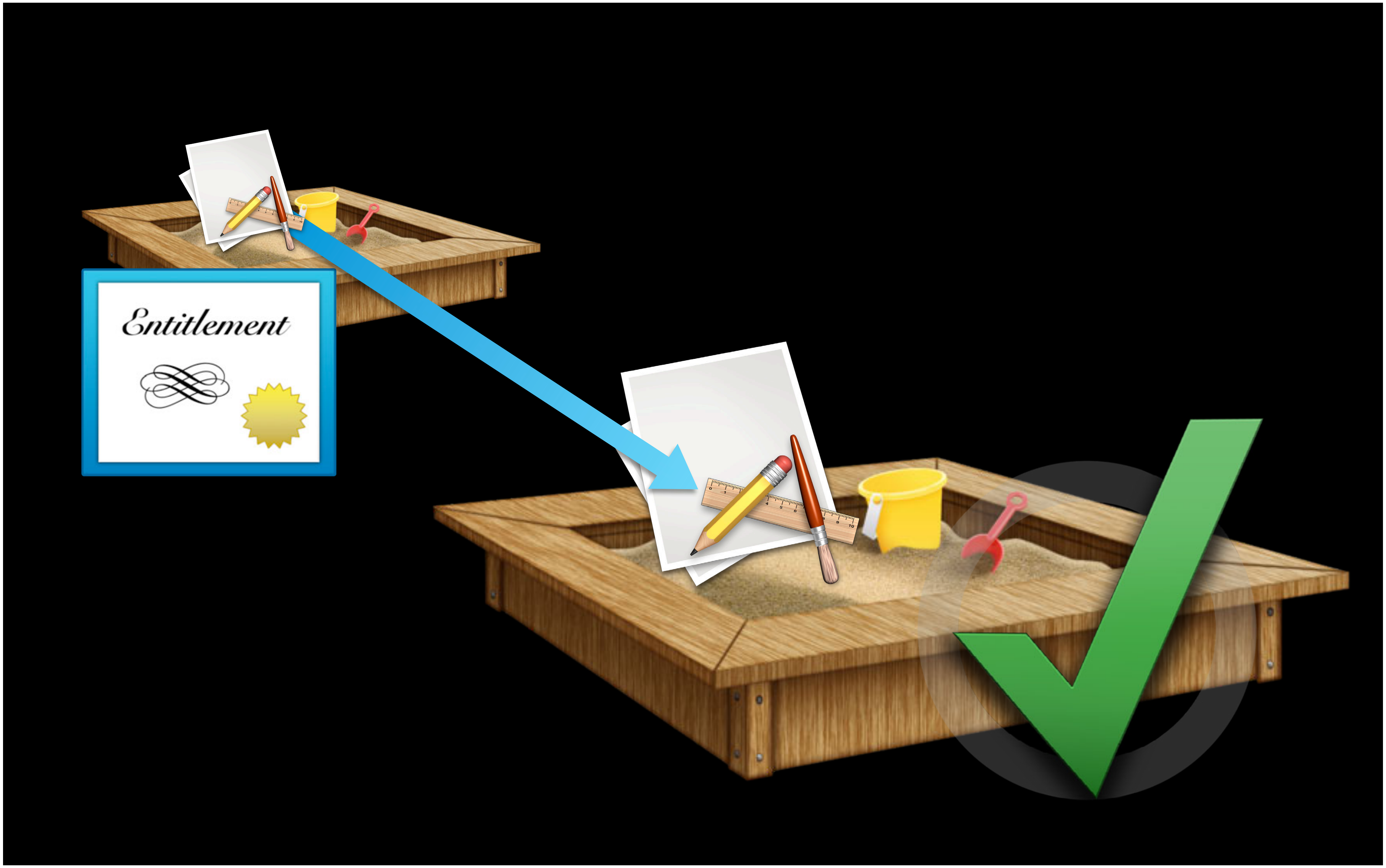
Apple events

Apple events

Apple events




Entitlement


Entitlement

Apple events and Entitlements

Chris Nebel
Senior Engineer Automation Technologies

Background

Apple event security policy

- No restrictions on receiving events
- Scriptable applications still scriptable when sandboxed
  - No code changes needed*

*File references must be type “file”, not type “text”.

Background

Apple event security policy

- Sending events is restricted
- One sandbox escape can ruin your whole day
- Apple events can escape sandbox
  - Use Finder to escape file system restrictions
  - Use Safari to escape network restrictions
  - Use Terminal to escape everything!
- Therefore, no Apple events by default
- Entitlement allows sending to a particular application

Apple events Entitlement

Sending events to Mail

```
<key>com.apple.security.temporary-exception.apple-events<key>
<array>
    <string>com.apple.mail<string>
</array>
```

Apple events Entitlement

Sending events to Mail

```
<key>com.apple.security.temporary-exception.apple-events<key>
<array>
    <string>com.apple.mail<string>
</array>
```

Apple events Entitlement

Sending events to Mail

```
<key>com.apple.security.temporary-exception.apple-events<key>
<array>
    <string>com.apple.mail<string>
</array>
```

Principle of Least Privilege

What are your intentions?

- Just enough privileges to do your job, and no more
- Existing entitlement is too broad
  - Grants complete access to an application
- Need ability to ask for just part of a scripting interface

Apple event Access Groups

Be specific!

- *Access groups* define groups of scriptable operations
  - Commands, classes, properties
  - Part of the application's scripting interface (sdef)
  - man 5 sdef
- Already in Mountain Lion applications
  - Mail: `com.apple.Mail.compose`
  - iTunes: `com.apple.iTunes.playback`, `com.apple.iTunes.library.read`, `com.apple.iTunes.library.read-write`

Defining an Access Group

Compose Mail message

```
<class-extension name="application">
    <element name="outgoing message"/>
</class>



<class name="outgoing message">
    ...
<class>


<command name="send">
    <direct-parameter type="outgoing message"/>
</command>
```

Defining an Access Group

Compose Mail message

```
<class-extension name="application">
    <element name="outgoing message">
        <access-group identifier="com.apple.Mail.compose" access="rw"/>
    </element>
</class>

<class name="outgoing message">
    ...
<class>


<command name="send">
    <direct-parameter type="outgoing message"/>
</command>
```

Defining an Access Group

Compose Mail message

```
<class-extension name="application">
    <element name="outgoing message">
        <access-group identifier="com.apple.Mail.compose" access="rw"/>
    </element>
</class>

<class name="outgoing message">
    <access-group identifier="com.apple.Mail.compose" access="rw"/>
     ...
<class>

<command name="send">
    <direct-parameter type="outgoing message"/>
</command>
```

Defining an Access Group

Compose Mail message

```
<class-extension name="application">
    <element name="outgoing message">
        <access-group identifier="com.apple.Mail.compose" access="rw"/>
    </element>
</class>

<class name="outgoing message">
    <access-group identifier="com.apple.Mail.compose" access="rw"/>
     ...
<class>

<command name="send">
    <!-- Not part of any access group.  No sending for you! -->
    <direct-parameter type="outgoing message"/>
</command>
```

Defining Access Groups

For your application

- Scriptable? Define access groups!
- Divide scripting interface along functional boundaries
  - Different kinds of clients
  - Different kinds of tasks
- Access groups may overlap
- Not everything needs to be in an access group

Using an Access Group

Be specific!

- New entitlement `com.apple.security.scripting-targets`
- Value is a dictionary
  - Keys are application code signing identifiers
  - Values are access group identifiers

Using an Access Group

Compose Mail message

```
<key>com.apple.security.scripting-targets<key>
<dict>
    <key>com.apple.Mail</key>
    <array>
        <string>com.apple.Mail.compose<string>
    </array>
</dict>
```

Using an Access Group

Compose Mail message

```
<key>com.apple.security.scripting-targets<key>
<dict>
    <key>com.apple.Mail</key>
    <array>
        <string>com.apple.Mail.compose<string>
    </array>
</dict>
```

Using an Access Group

Compose Mail message

```
<key>com.apple.security.scripting-targets<key>
<dict>
    <key>com.apple.Mail</key>
    <array>
        <string>com.apple.Mail.compose<string>
    </array>
</dict>
```

Using an Access Group

Compose Mail message

```
<key>com.apple.security.scripting-targets<key>
<dict>
    <key>com.apple.Mail</key>
    <array>
        <string>com.apple.Mail.compose<string>
    </array>
</dict>
```

App-to-App Automation

Using Apple events with your application

- Receiving Apple events?
  - Keep existing code
  - Add access groups
- Sending Apple events?
  - Keep existing code, but sandboxed apps need an entitlement
  - Use `com.apple.security.scripting-targets` if you can
  - Otherwise, use `com.apple.security.temporary-exception.apple-events`

Apple events and Entitlements

Chris Nebel
Senior Engineer Automation Technologies

Attaching Scripts

Running user scripts in your application

Application-Run User Scripts

Application-Run User Scripts

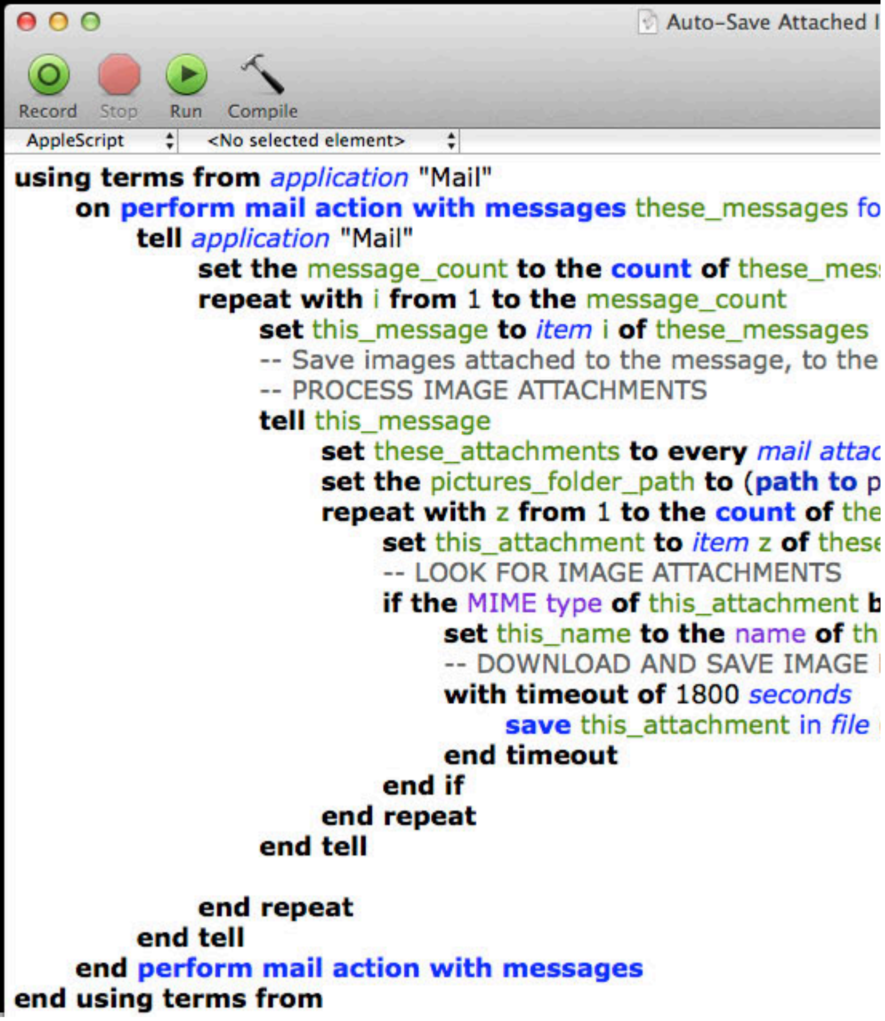
- Application Script Menu

Application-Run User Scripts

- Application Script Menu
- Event Handlers

Application-Run User Scripts

- Application Script Menu
- Event Handlers
  - Mail Rule

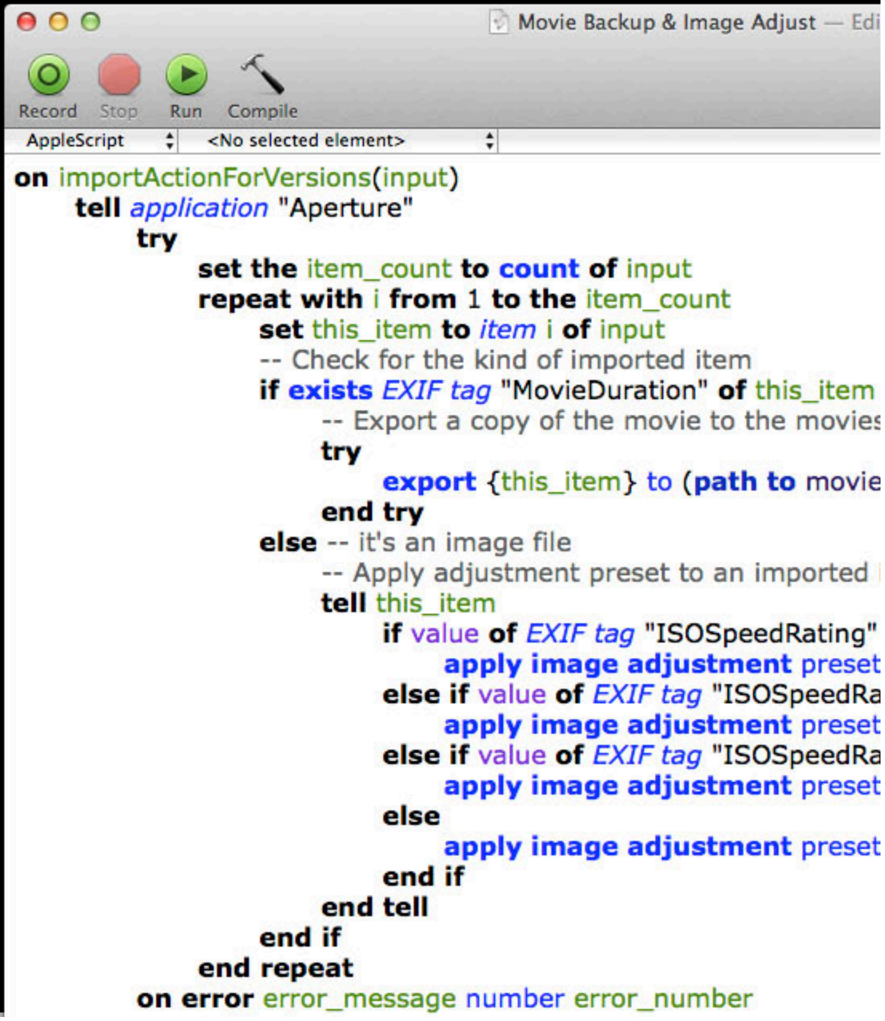


```
using terms from application "Mail"
    on perform mail action with messages these_messages fo
        tell application "Mail"
            set the message_count to the count of these_mes
            repeat with i from 1 to the message_count
                set this_message to item i of these_messages
                -- Save images attached to the message, to the
                -- PROCESS IMAGE ATTACHMENTS
                tell this_message
                    set these_attachments to every mail attac
                    set the pictures_folder_path to (path to p
                    repeat with z from 1 to the count of the
                        set this_attachment to item z of these
                        -- LOOK FOR IMAGE ATTACHMENTS
                        if the MIME type of this_attachment b
                            set this_name to the name of th
                            -- DOWNLOAD AND SAVE IMAGE
                            with timeout of 1800 seconds
                                save this_attachment in file
                            end timeout
                        end if
                    end repeat
                end tell

            end repeat
        end tell
    end perform mail action with messages
end using terms from
```

Application-Run User Scripts

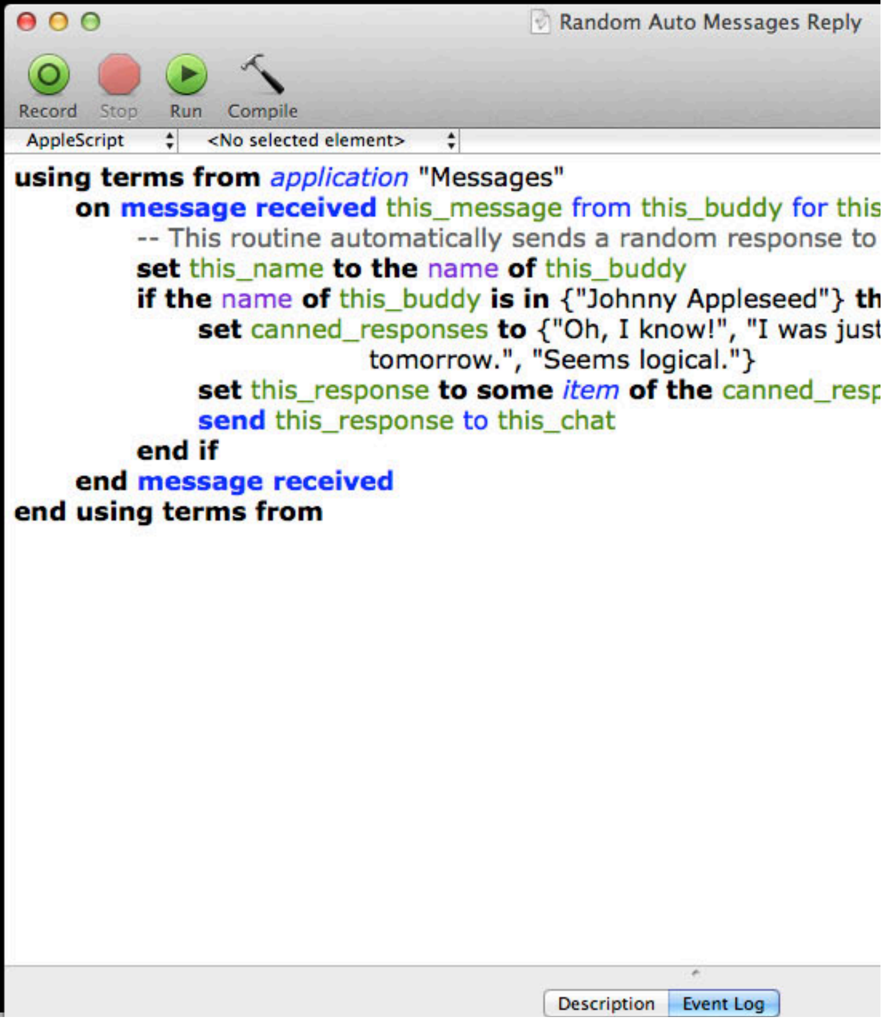
- Application Script Menu
- Event Handlers
  - Mail Rule
  - Aperture Import Action



```
on importActionForVersions(input)
    tell application "Aperture"
        try
            set the item_count to count of input
            repeat with i from 1 to the item_count
                set this_item to item i of input
                -- Check for the kind of imported item
                if exists EXIF tag "MovieDuration" of this_item
                    -- Export a copy of the movie to the movies
                    try
                        export {this_item} to (path to movie
                    end try
                else -- it's an image file
                    -- Apply adjustment preset to an imported
                    tell this_item
                        if value of EXIF tag "ISOSpeedRating"
                            apply image adjustment preset
                        else if value of EXIF tag "ISOSpeedRa
                            apply image adjustment preset
                        else if value of EXIF tag "ISOSpeedRa
                            apply image adjustment preset
                        else
                            apply image adjustment preset
                        end if
                    end tell
                end if
            end repeat
        on error error_message number error_number
```

Application-Run User Scripts

- Application Script Menu
- Event Handlers
  - Mail Rule
  - Aperture Import Action
  - Messages Events



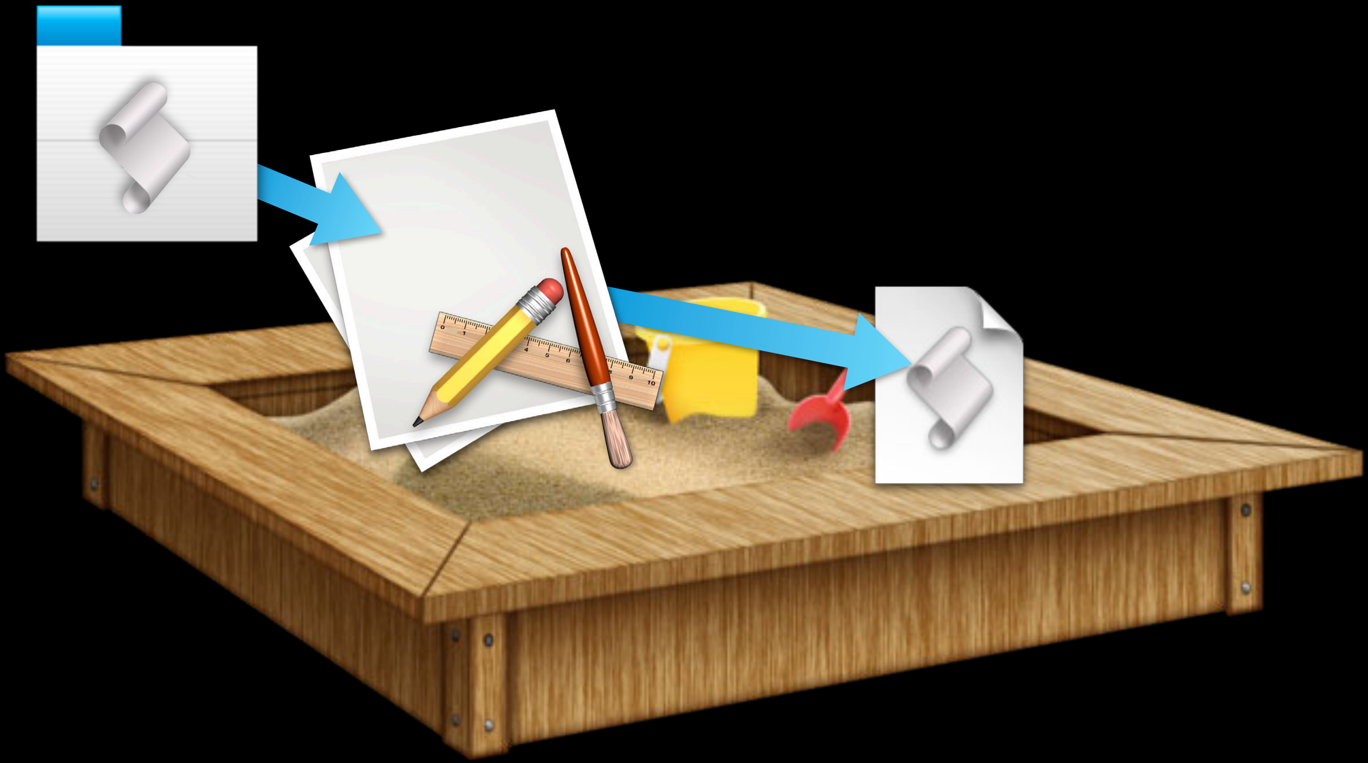
```
using terms from application "Messages"
	on message received this_message from this_buddy for this
		-- This routine automatically sends a random response to
		set this_name to the name of this_buddy
		if the name of this_buddy is in {"Johnny Appleseed"} th
			set canned_responses to {"Oh, I know!", "I was just
				tomorrow.", "Seems logical."}
			set this_response to some item of the canned_resp
			send this_response to this_chat
		end if
	end message received
end using terms from
```

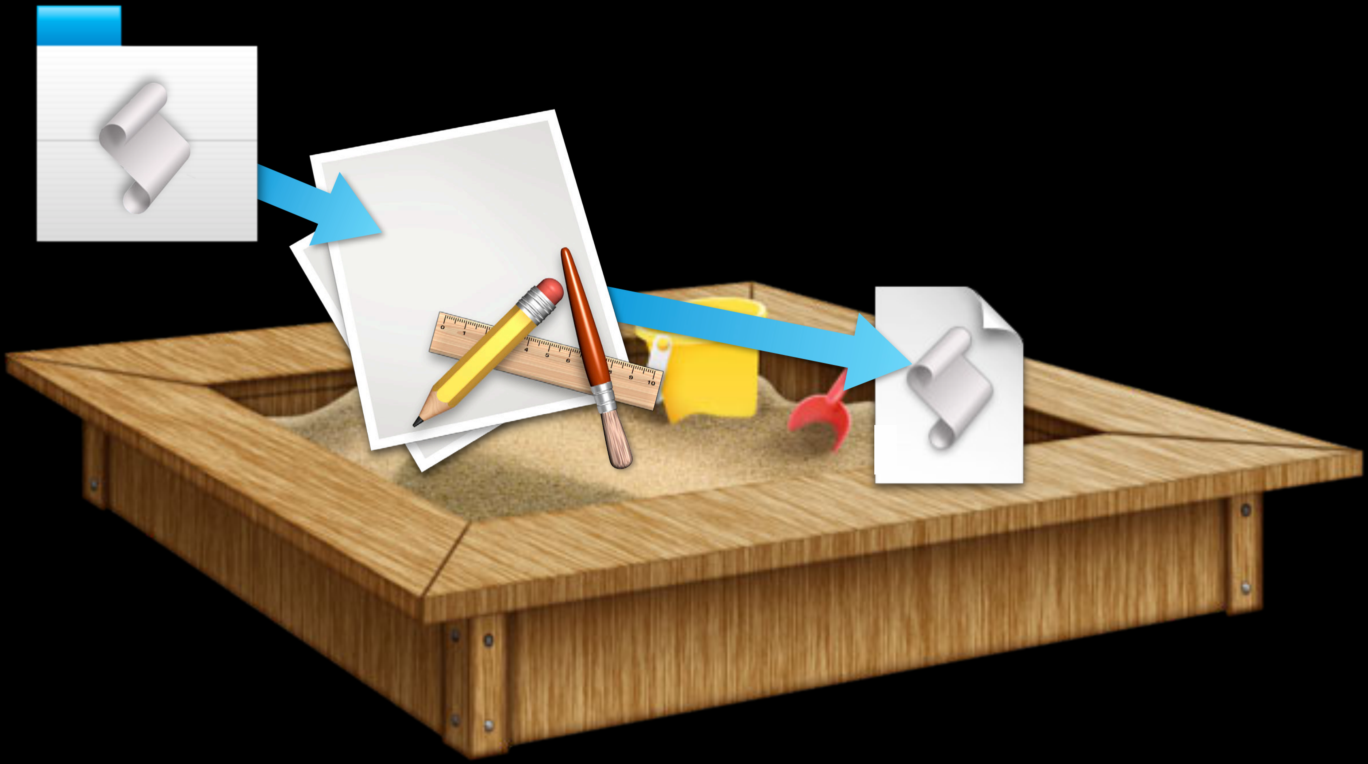
Application-Run User Scripts

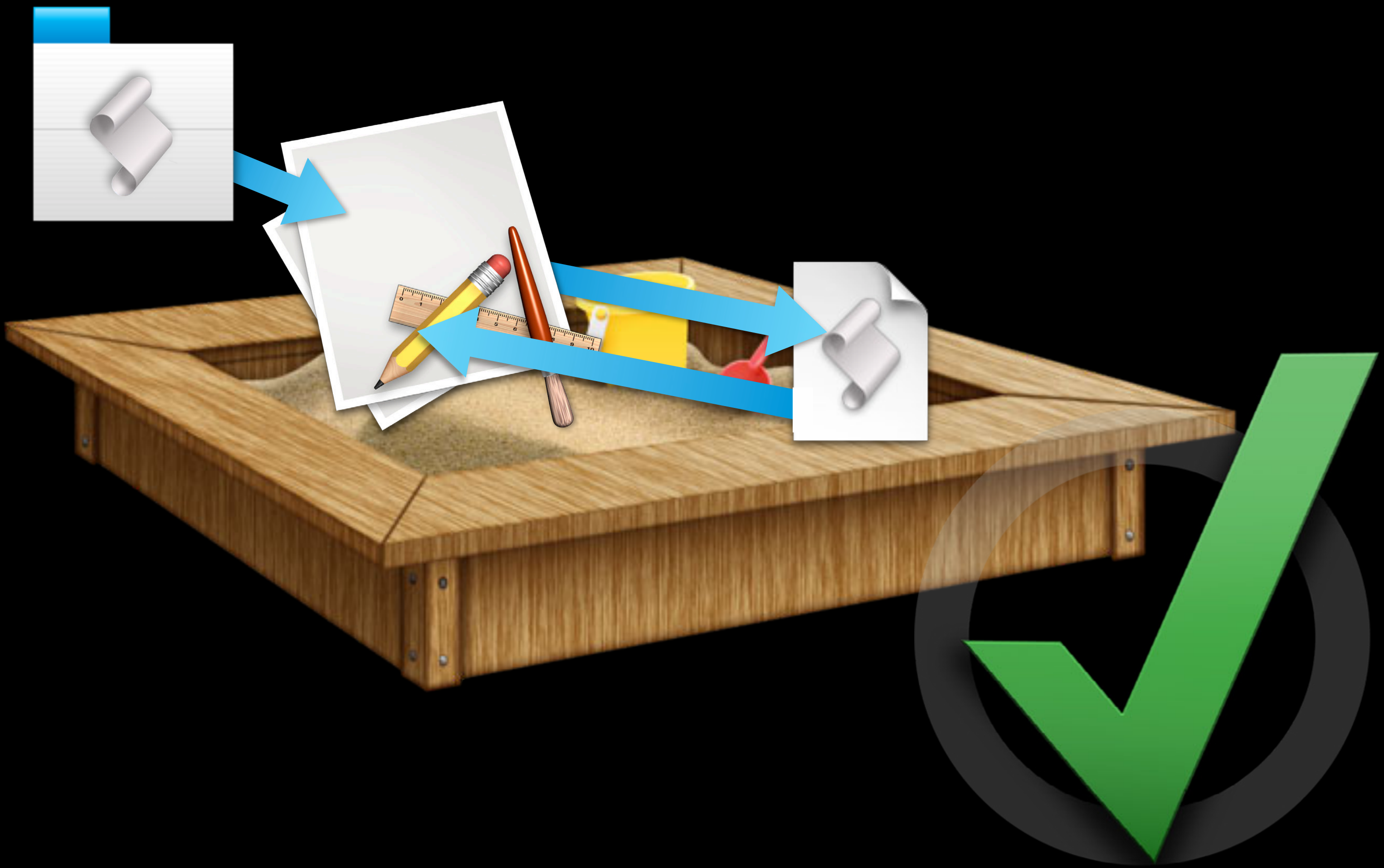
- Application Script Menu
- Event Handlers
  - Mail Rule
  - Aperture Import Action
  - Messages Events
- Scripts executed by hosting application

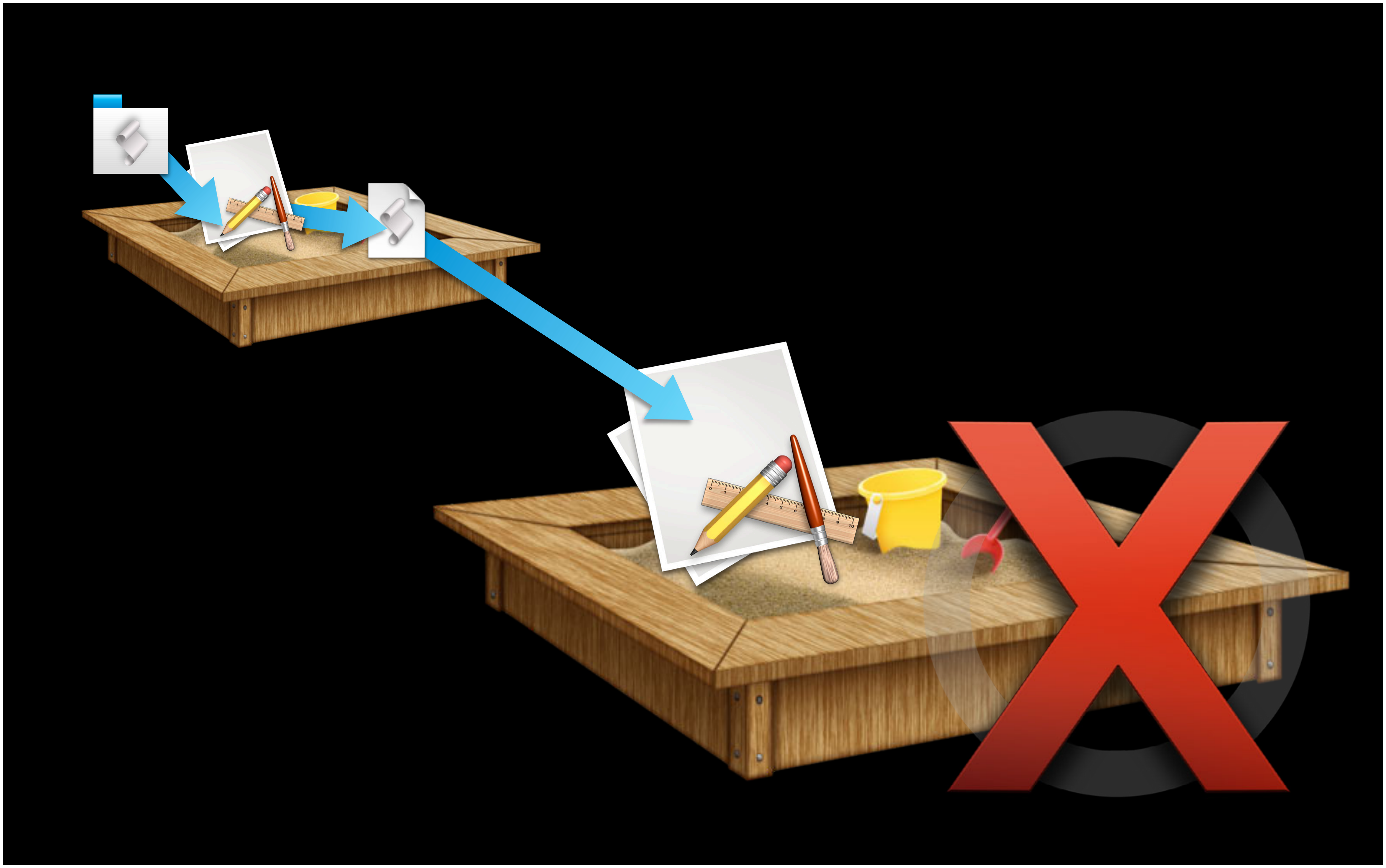
Application-Run User Scripts

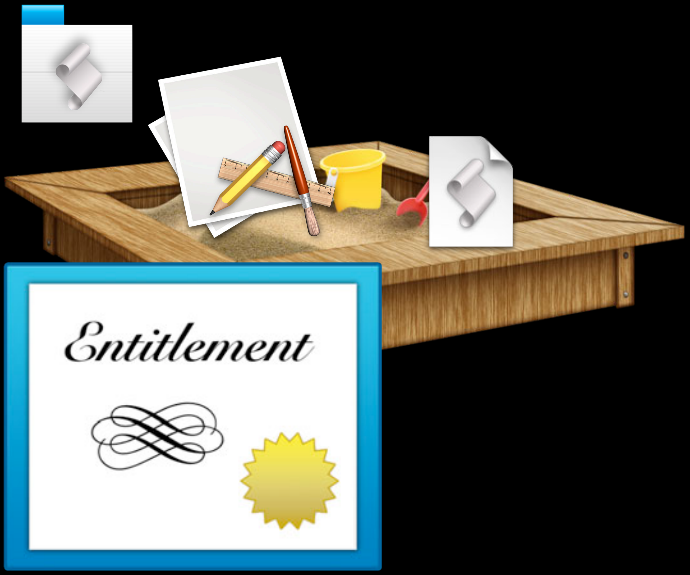
- Application Script Menu
- Event Handlers
  - Mail Rule
  - Aperture Import Action
  - Messages Events
- Scripts executed by hosting application
- Inherit application’s permissions

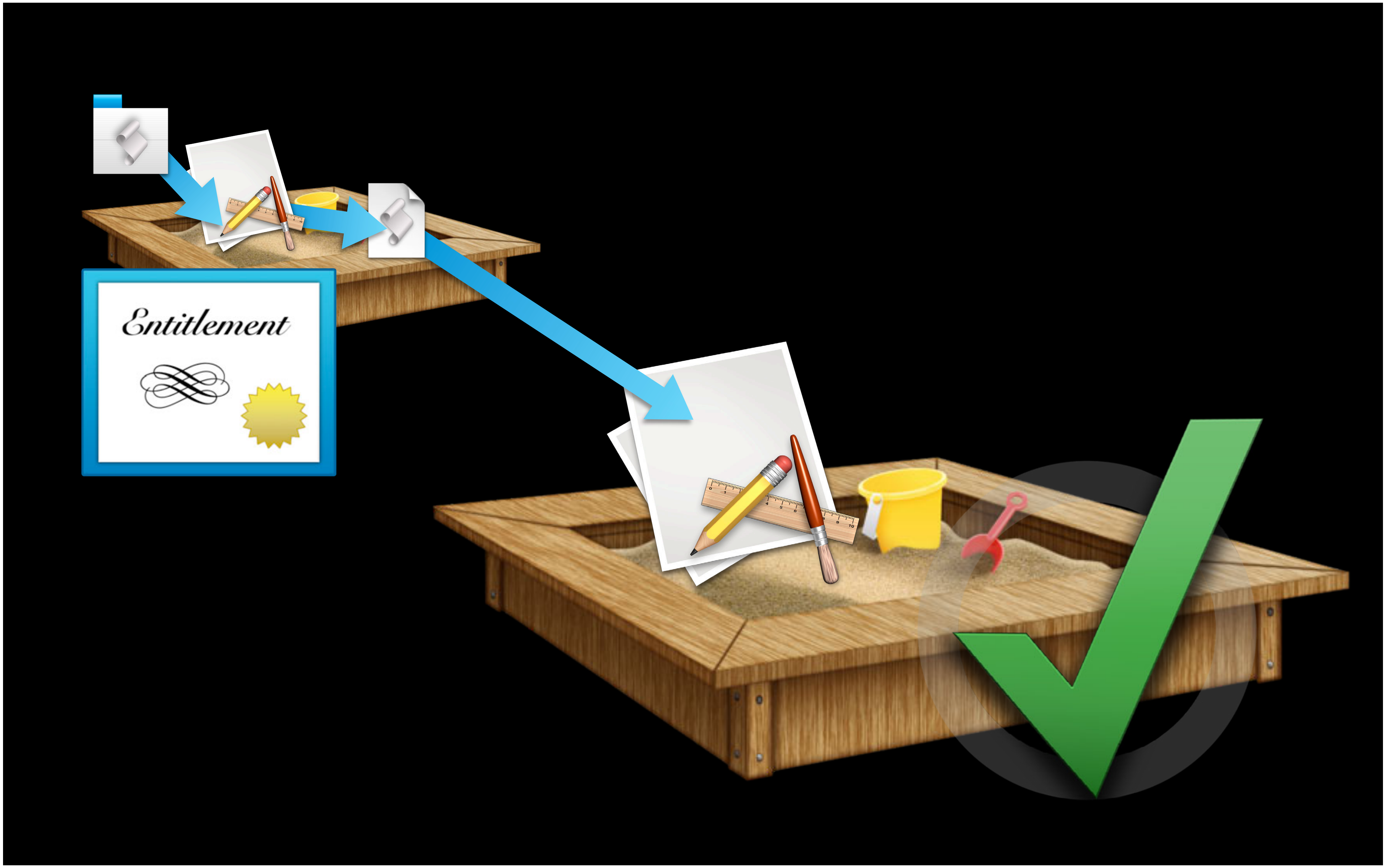


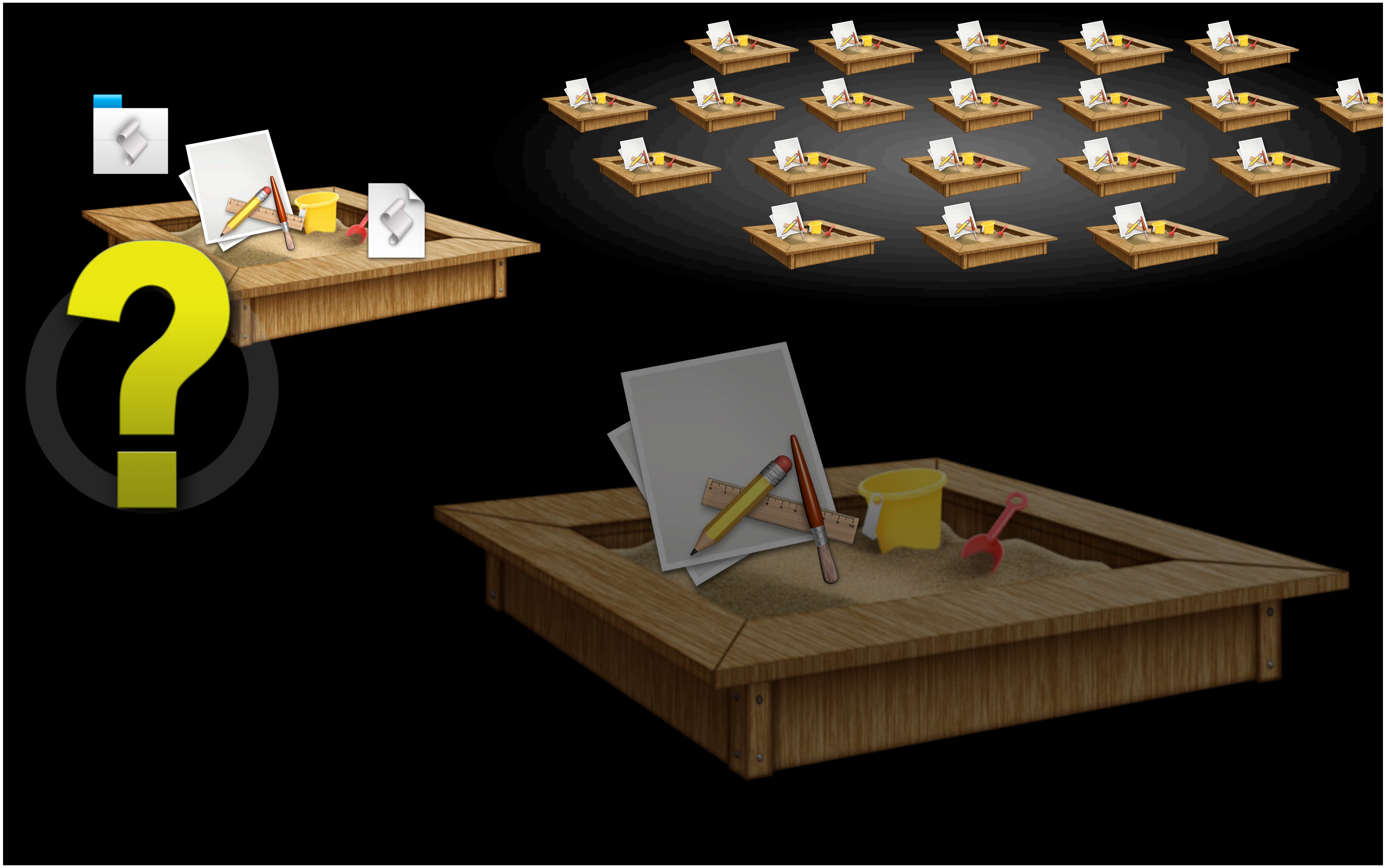




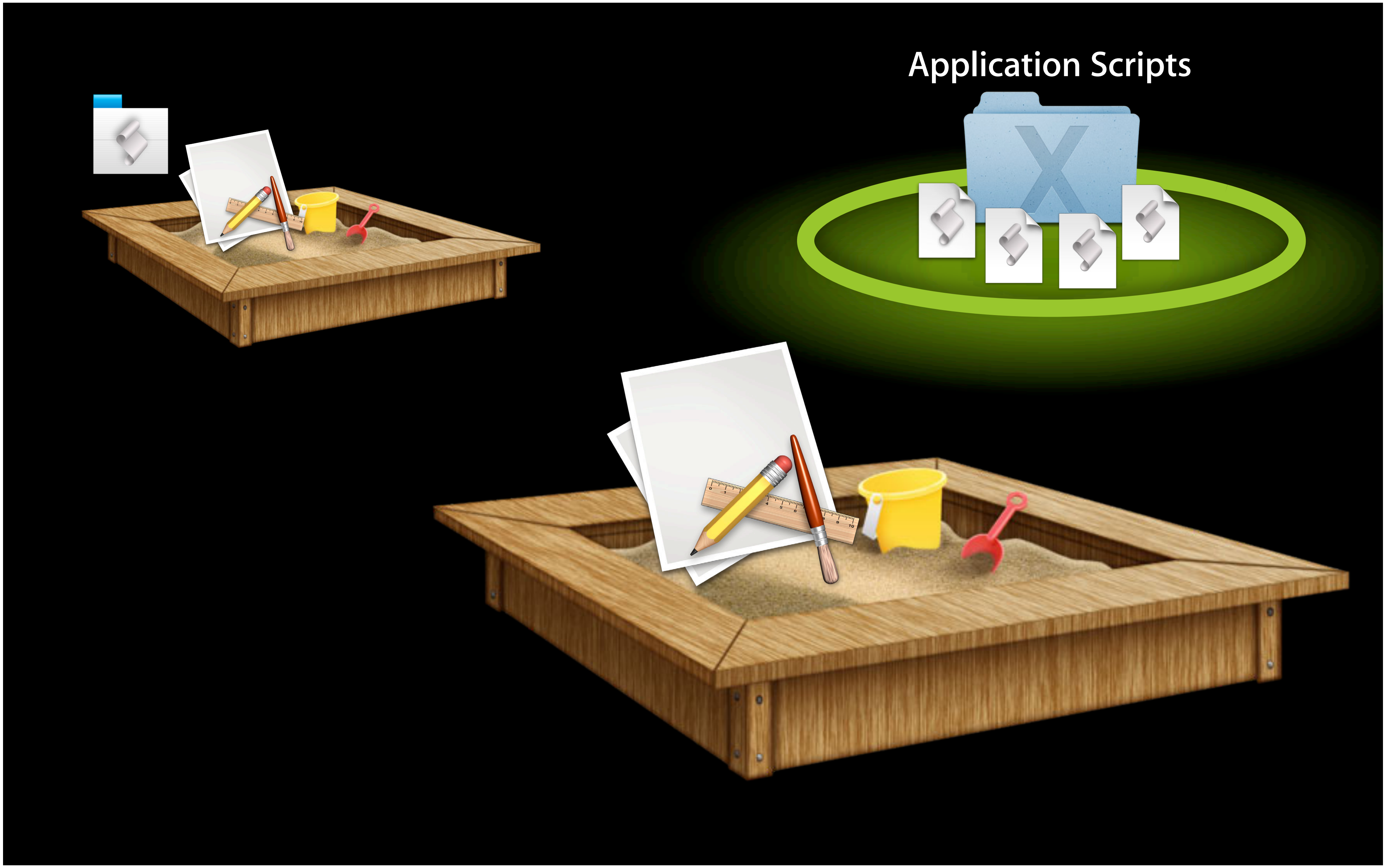


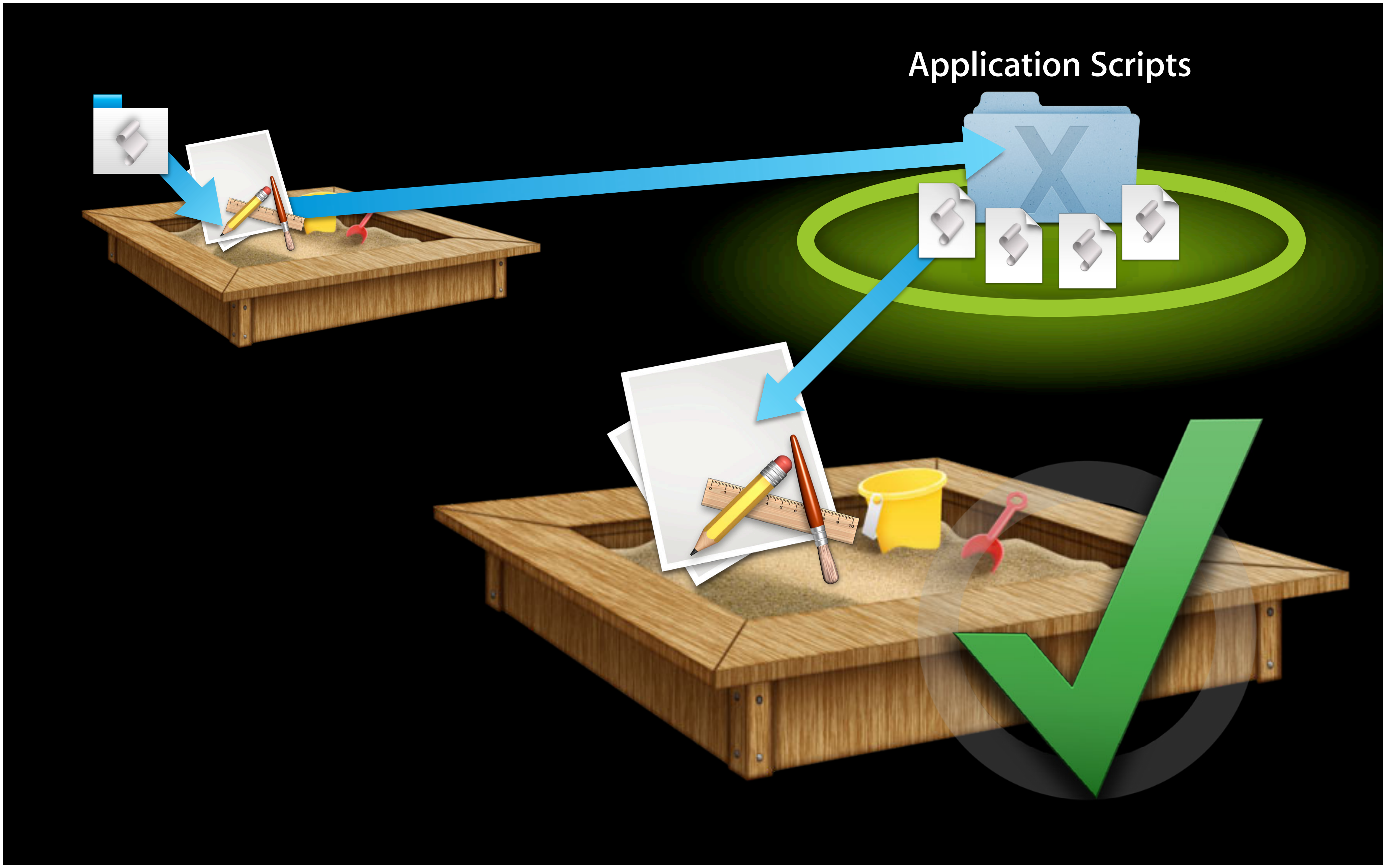

Entitlement

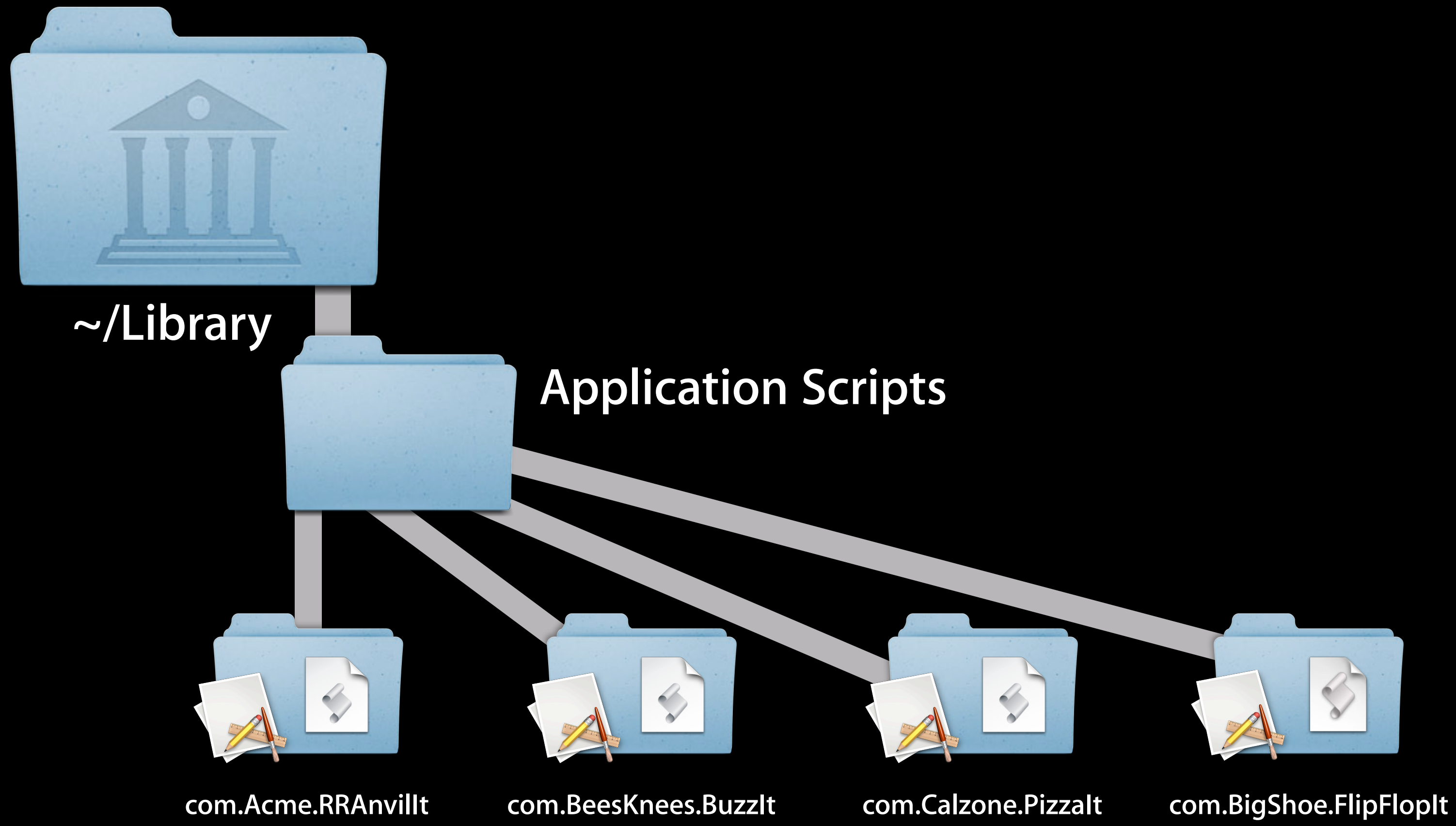

Entitlement

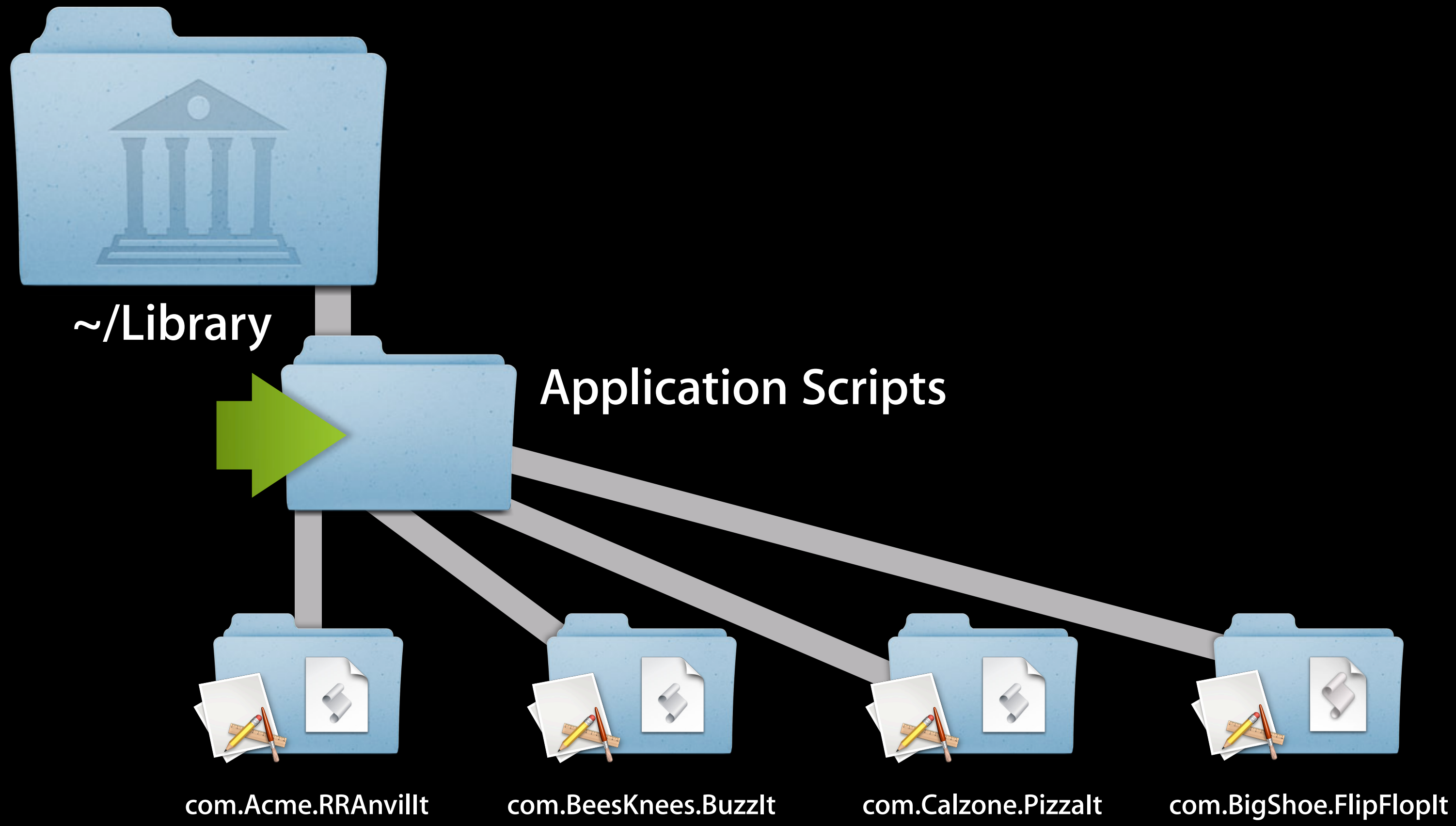


Application Scripts

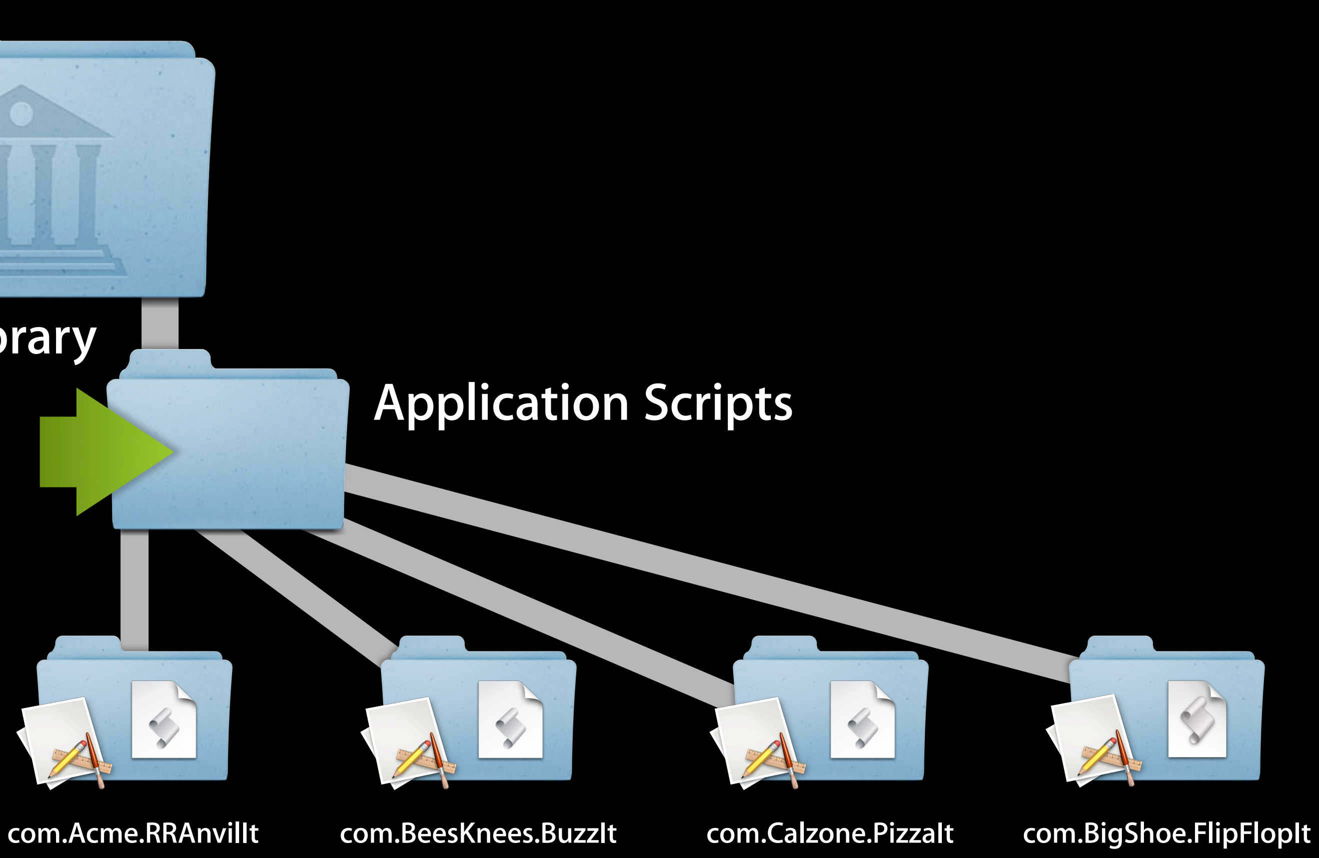

Application Scripts


Application Scripts


~/Library
Application Scripts
com.Acme.RRAnvilIt
com.BeesKnees.BuzzIt
com.Calzone.PizzaIt
com.BigShoe.FlipFlopIt


~/Library
Application Scripts
com.Acme.RRAnvilIt
com.BeesKnees.BuzzIt
com.Calzone.PizzaIt
com.BigShoe.FlipFlopIt

NSUserScriptTask



Attaching Scripts

Running user scripts in your application

Chris Nebel
Senior Engineer Automation Technologies

NSUserScriptTask

Running user scripts

- Unified class for running user-supplied scripts
- NSUserScriptTask for generic scripts
  - Supports AppleScript, Automator, and Unix scripts
- Subclasses for specific control
  - NSUserAppleScriptTask, NSUserAutomatorTask, NSUserUnixTask
- Part of Foundation.framework

NSUserScriptTask

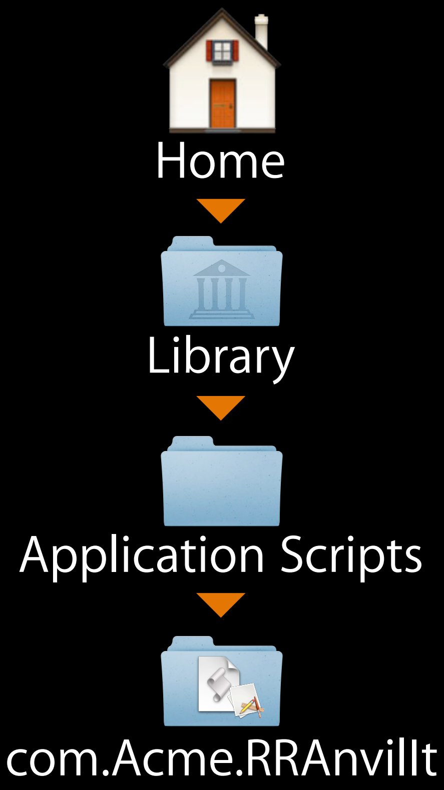
If sandboxed…

- Script runs outside the sandbox
- Scripts must be in blessed folder

```
NSApplicationScriptsDirectory

~/Library/Application Scripts/code-signing-identifier
```

- Application may read from, but not write to, blessed folder
- Folder sub-structure is up to the application
- No entitlement required



NSUserScriptTask

Locating the scripts folder

```
NSURL *scriptsFolderURL;
NSError *error;

scriptsFolderURL = [[NSFileManager defaultManager]
      URLForDirectory:NSApplicationScriptsDirectory
             inDomain:NSUserDomainMask
    appropriateForURL:nil
               create:YES
                error:&error];
```

NSUserScriptTask

Locating the scripts folder

```
NSURL *scriptsFolderURL;
NSError *error;

scriptsFolderURL = [[NSFileManager defaultManager]
      URLForDirectory:NSApplicationScriptsDirectory
             inDomain:NSUserDomainMask
    appropriateForURL:nil
               create:YES
                error:&error];
```

NSUserScriptTask

Running a script

```
NSUserScriptTask *script;
NSError *error;

script = [[NSUserScriptTask alloc] initWithURL:scriptURL error:&error];

[script executeWithCompletionHandler:
    ^(NSError *error) {
        if (!error) {
            // Success!
        } else {
            NSLog(@"error: %@", error);
        }
    }
];
```

NSUserScriptTask

Running a script

```
NSUserScriptTask *script;
NSError *error;

script = [[NSUserScriptTask alloc] initWithURL:scriptURL error:&error];

[script executeWithCompletionHandler:
    ^(NSError *error) {
        if (!error) {
            // Success!
        } else {
            NSLog(@"error: %@", error);
        }
    }
];
```

NSUserScriptTask

Running a script

```
NSUserScriptTask *script;
NSError *error;

script = [[NSUserScriptTask alloc] initWithURL:scriptURL error:&error];

[script executeWithCompletionHandler:
    ^(NSError *error) {
        if (!error) {
            // Success!
        } else {
            NSLog(@"error: %@", error);
        }
    }
];
```

NSUserScriptTask

Running an AppleScript script

```
NSUserAppleScriptTask *script;
NSError *error;

script = [[NSUserAppleScriptTask alloc] initWithURL:scriptURL error:&error];

NSAppleEventDescriptor *event = [self newHandlerEvent];

[script executeWithAppleEvent:event completionHandler:
    ^(NSAppleEventDescriptor *result, NSError *error) {
        if (!error) {
            // Success!
        } else {
            NSLog(@"error: %@", error);
        }
    }
];
```

NSUserScriptTask

Running an AppleScript script

```
NSUserAppleScriptTask *script;
NSError *error;

script = [[NSUserAppleScriptTask alloc] initWithURL:scriptURL error:&error];

NSAppleEventDescriptor *event = [self newHandlerEvent];

[script executeWithAppleEvent:event completionHandler:
    ^(NSAppleEventDescriptor *result, NSError *error) {
        if (!error) {
            // Success!
        } else {
            NSLog(@"error: %@", error);
        }
    }
];
```

NSUserScriptTask

Running an AppleScript script

```
NSUserAppleScriptTask *script;
NSError *error;

script = [[NSUserAppleScriptTask alloc] initWithURL:scriptURL error:&error];

NSAppleEventDescriptor *event = [self newHandlerEvent];

[script executeWithAppleEvent:event completionHandler:
    ^(NSAppleEventDescriptor *result, NSError *error) {
        if (!error) {
            // Success!
        } else {
            NSLog(@"error: %@", error);
        }
    }
];
```

NSUserScriptTask

Running an AppleScript script

```
NSUserAppleScriptTask *script;
NSError *error;

script = [[NSUserAppleScriptTask alloc] initWithURL:scriptURL error:&error];

NSAppleEventDescriptor *event = [self newHandlerEvent];

[script executeWithAppleEvent:event completionHandler:
    ^(NSAppleEventDescriptor *result, NSError *error) {
        if (!error) {
            // Success!
        } else {
            NSLog(@"error: %@", error);
        }
    }
];
```

NSUserScriptTask

Running an AppleScript script

```
NSUserAppleScriptTask *script;
NSError *error;

script = [[NSUserAppleScriptTask alloc] initWithURL:scriptURL error:&error];

NSAppleEventDescriptor *event = [self newHandlerEvent];

[script executeWithAppleEvent:event completionHandler:
    ^(NSAppleEventDescriptor *result, NSError *error) {
        if (!error) {
            // Success!
        } else {
            NSLog(@"error: %@", error);
        }
    }
];
```

Attaching Scripts

Running user scripts in your application

- Use `NSUserScriptTask` to run user scripts
  - For user-supplied scripts only
  - Can replace `NSAppleScript`, `AMWorkflow`, `NSTask`
- Use specific subclasses for special control
- Use application scripts folder for scripts
  - Script must be in application scripts folder if sandboxed

Attaching Scripts

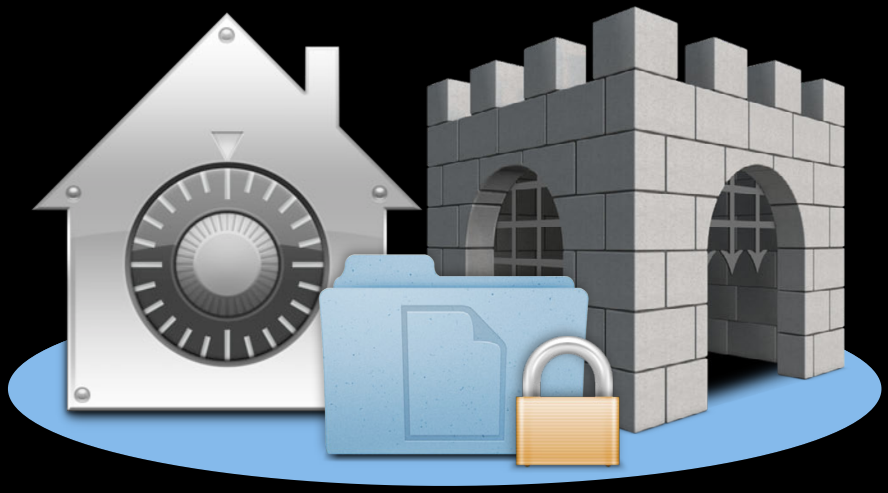
Running user scripts in your application

Chris Nebel
Senior Engineer Automation Technologies

Session Summary

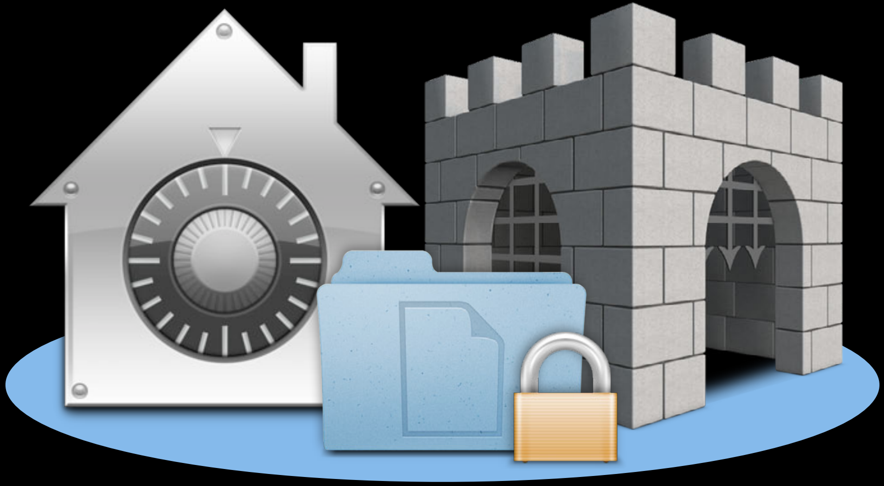
Session Summary

- Security is a focus in Mountain Lion



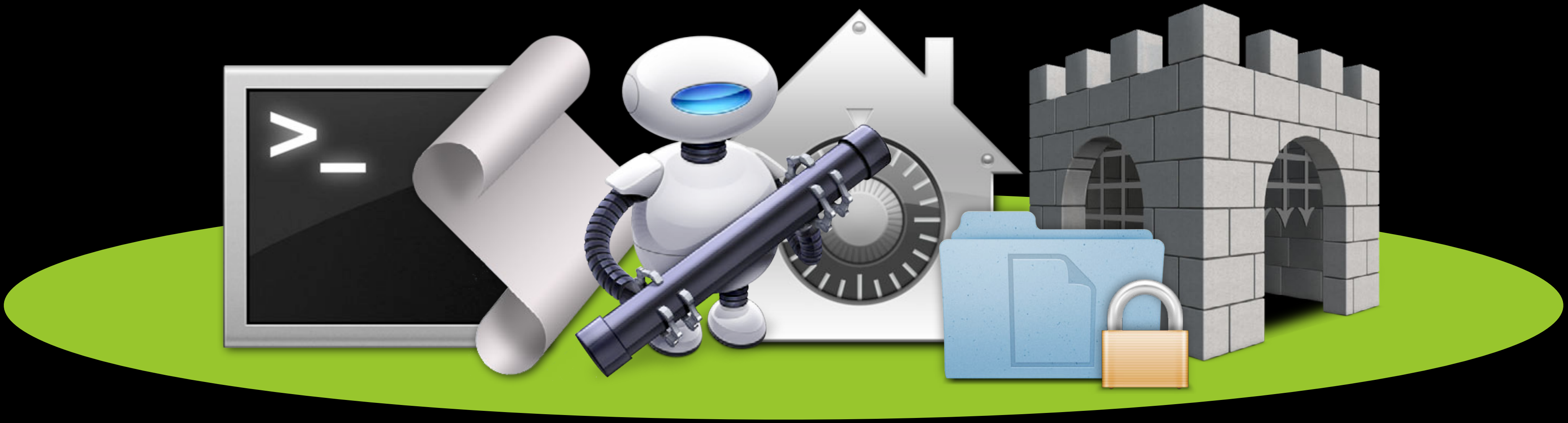
Session Summary

- Security is a focus in Mountain Lion
  - Automation remains an essential element of OS X



Session Summary

- Security is a focus in Mountain Lion
  - Automation remains an essential element of OS X



Session Summary

- Security is a focus in Mountain Lion
    - Automation remains an essential element of OS X

Session Summary

- Security is a focus in Mountain Lion
    - Automation remains an essential element of OS X
- Personal automation

Session Summary

- Security is a focus in Mountain Lion
    - Automation remains an essential element of OS X
- Personal automation
    - No changes, no restrictions

Session Summary

- Security is a focus in Mountain Lion
  - Automation remains an essential element of OS X
- Personal automation
  - No changes, no restrictions
- Distributing scripts

Session Summary

- Security is a focus in Mountain Lion
  - Automation remains an essential element of OS X
- Personal automation
  - No changes, no restrictions
- Distributing scripts
  - Sign applets to work with Gatekeeper

Session Summary

- Security is a focus in Mountain Lion
  - Automation remains an essential element of OS X
- Personal automation
  - No changes, no restrictions
- Distributing scripts
  - Sign applets to work with Gatekeeper
- Application-to-application

Session Summary

- Security is a focus in Mountain Lion
    - Automation remains an essential element of OS X
- Personal automation
    - No changes, no restrictions
- Distributing scripts
    - Sign applets to work with Gatekeeper
- Application-to-application
    - Use entitlements with Apple event access groups

Session Summary

- Security is a focus in Mountain Lion
  - Automation remains an essential element of OS X
- Personal automation
  - No changes, no restrictions
- Distributing scripts
  - Sign applets to work with Gatekeeper
- Application-to-application
  - Use entitlements with Apple event access groups
- Attached scripts

Session Summary

- Security is a focus in Mountain Lion
    - Automation remains an essential element of OS X
- Personal automation
    - No changes, no restrictions
- Distributing scripts
    - Sign applets to work with Gatekeeper
- Application-to-application
    - Use entitlements with Apple event access groups
- Attached scripts
    - Use `NSUserScriptTask`

More Information

Michael Jurewitz
Developer Tools and Frameworks Evangelist
jury@apple.com

Documentation
App Sandboxing
http://developer.apple.com/devcenter/mac/app-sandbox/

Apple Developer Forums
http://devforums.apple.com

Related Sessions

The OS X App Sandbox	Nob Hill Friday 10:15AM
Gatekeeper and Developer ID	Nob Hill Tuesday 11:30AM

Labs

Automation Lab	Essentials Lab A Tuesday 4:30PM
Security Lab	Core OS Lab B Tuesday 3:15PM

WWDC2012