

# **The Past, Present, And Future of Extensions Management**

Charles Edge

# Part I



SECRET CHEST 17

### Discover Secret Chest



Discover the possibilities and see how easy it is to manage passwords with the most secure encryption available today.

Learn More



## Secrets

Search



Add Secret



See All

## Devices



Manage  
Devices

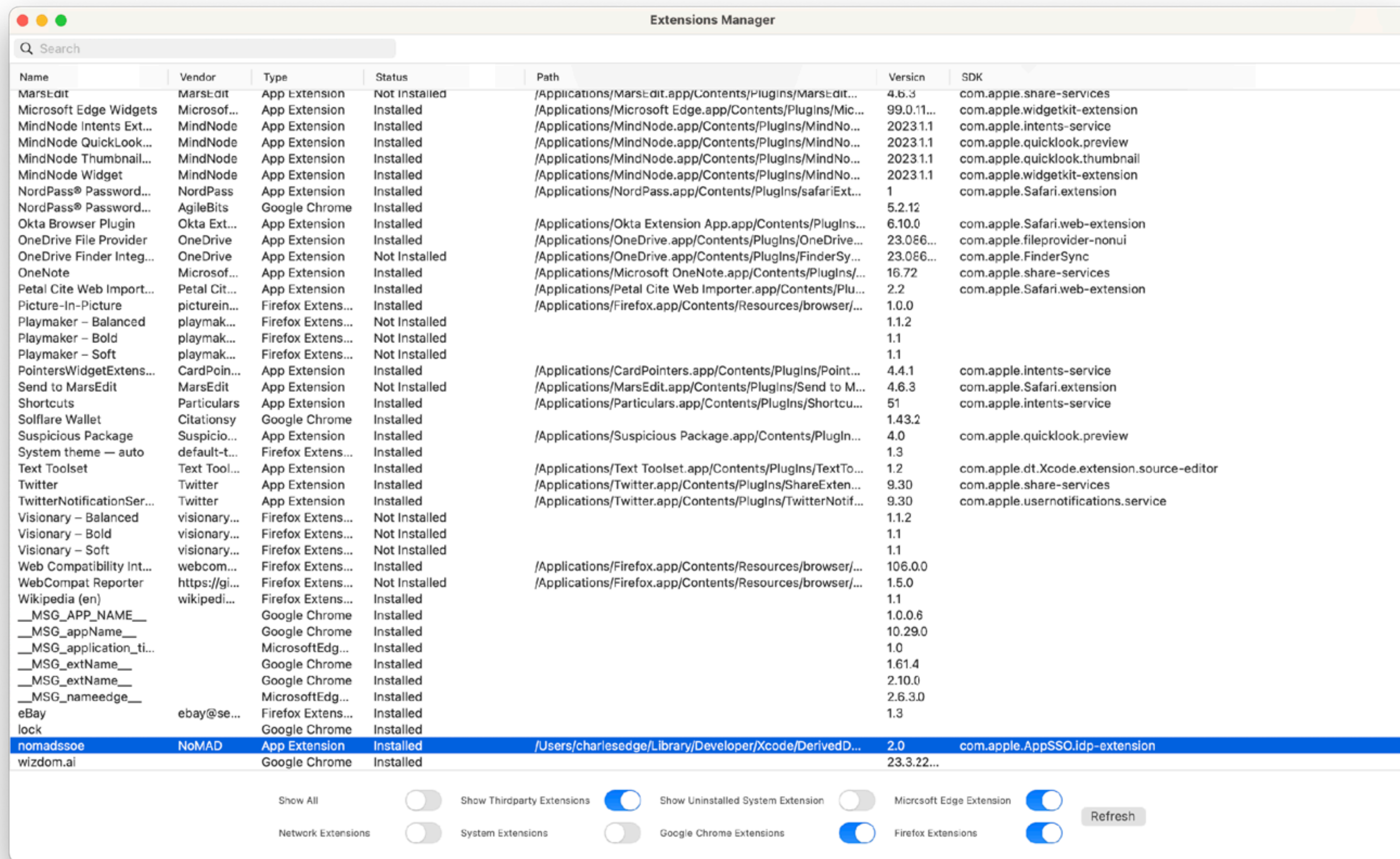


iPhone



CE MacBook  
Pro

# SecretChest.io



# The Return Of Extensions Manager

<https://github.com/krypted/extensionsmanager>



# Importing Code Into Swift

- Swift Package: Uncompiled folders and files with a specific format and a manifest.
- Library: Code and compiled code that can be imported into a project.
- xcframeworks: Opinionated, compiled code that has private and public structures.
- App Extensions: Discrete, signed, and compiled/factored pieces of code.
- Apps: Completed/compiled products available for distribution.
- SDK: Collection of artifacts (libraries, frameworks, & documentation) for types of applications.
- Kit: Apple's manifestation of an SDK.
- Extension: Keyword that extends a type with components, methods, types, etc.

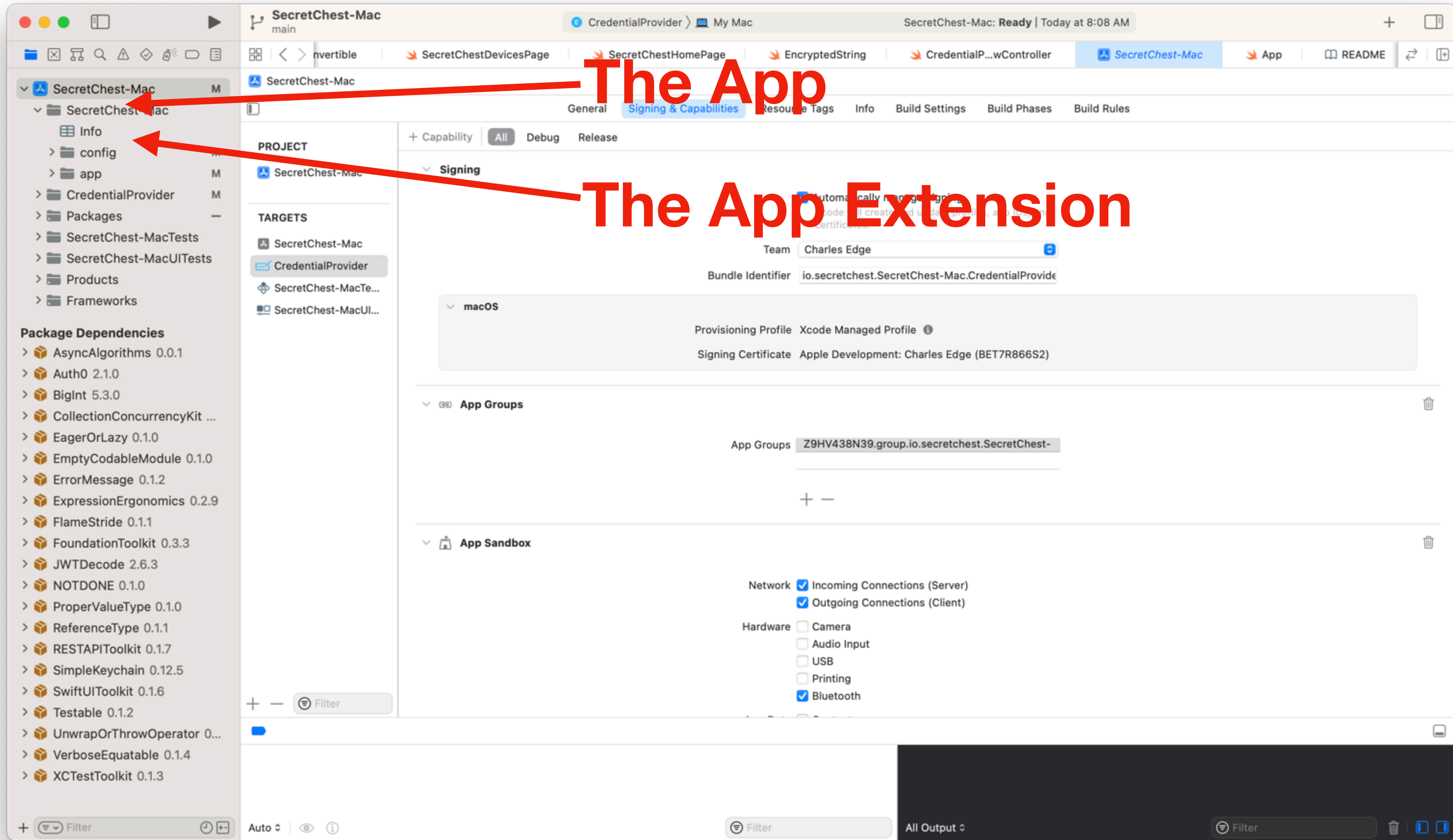
[https://www.secret-chest.com/post/  
design-patterns-and-technical-  
options-when-re-using-swift-code](https://www.secret-chest.com/post/design-patterns-and-technical-options-when-re-using-swift-code)

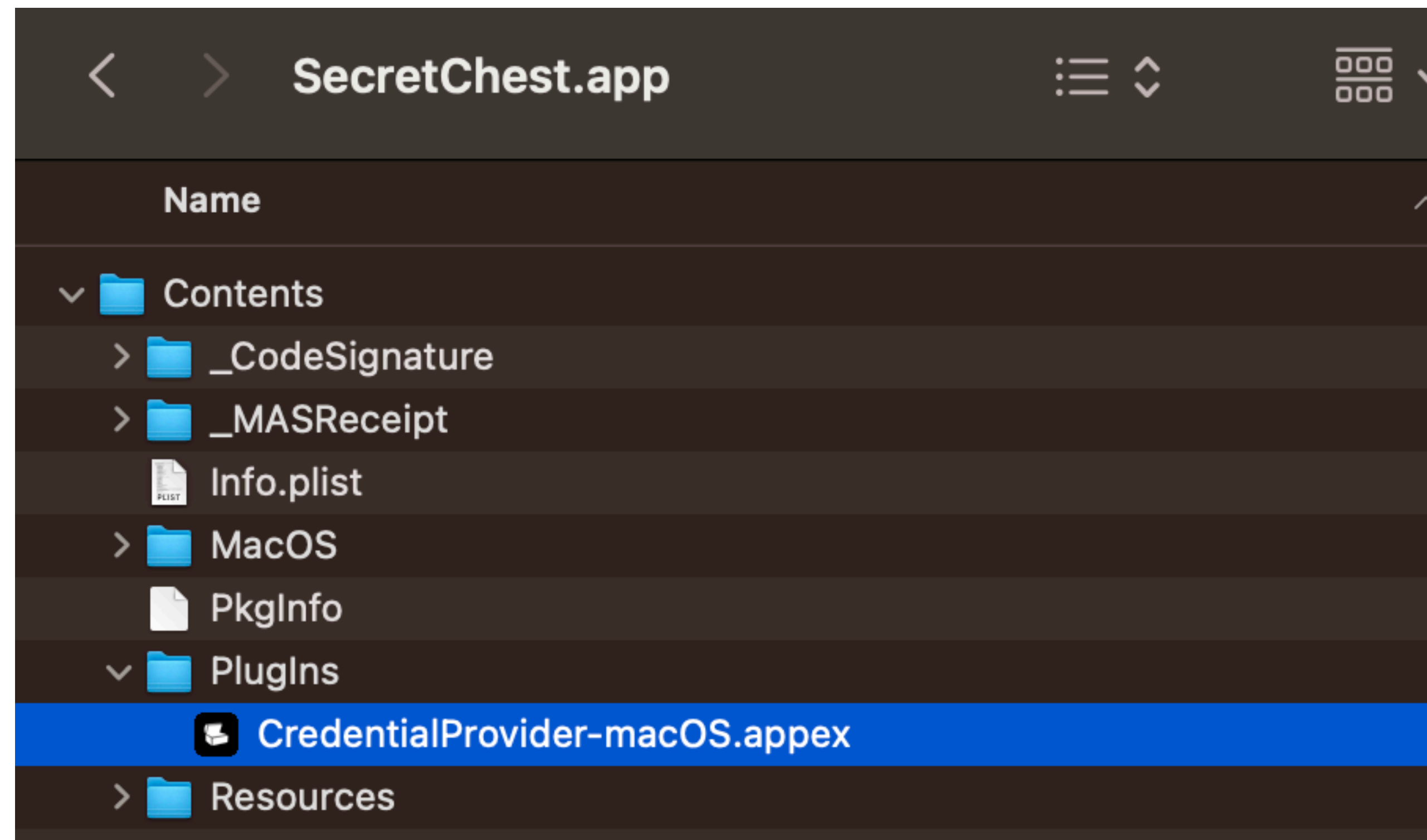


**Some extensions are from  
Apple, to be consumed**

**For those developers write, users  
just see the Apps and App  
Extensions in final products**









Good afternoon

Sign on to manage your accounts.

Username



SecretChest...  
iCloud Keychain...

Sign On

[Forgot username or password?](#)  
[Security Center](#)  
[Privacy, Cookies, and Legal](#)

CredentialProvider

Cancel

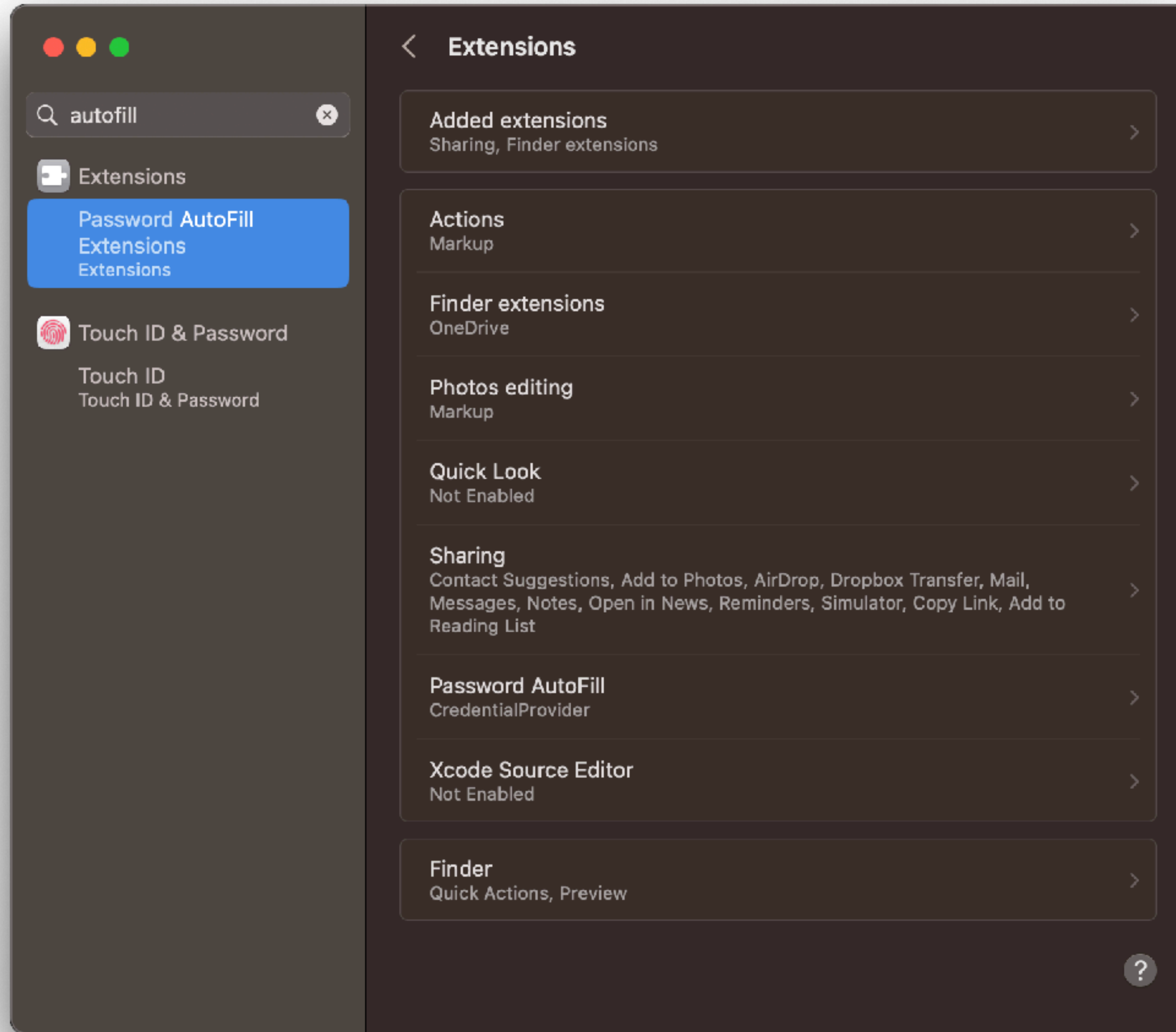
Return Example Password

0% intro APR for 21 months

with \$0 annual fee. Terms apply.

Learn more





🔍 autofill ✕

🔧 Extensions

**Password AutoFill**  
Extensions  
Extensions

👉 Touch ID & Password

Touch ID  
Touch ID & Password

## < Extensions

### Added extensions

Sharing, Finder extensions



### Actions

Markup



### Finder extensions

OneDrive



### Photos editing

Markup



### Quick Look

Not Enabled



### Sharing

Contact Suggestions, Add to Photos, AirDrop, Dropbox Transfer, Mail, Messages, Notes, Open in News, Reminders, Simulator, Copy Link, Add to Reading List



### Password AutoFill

CredentialProvider



### Xcode Source Editor

Not Enabled

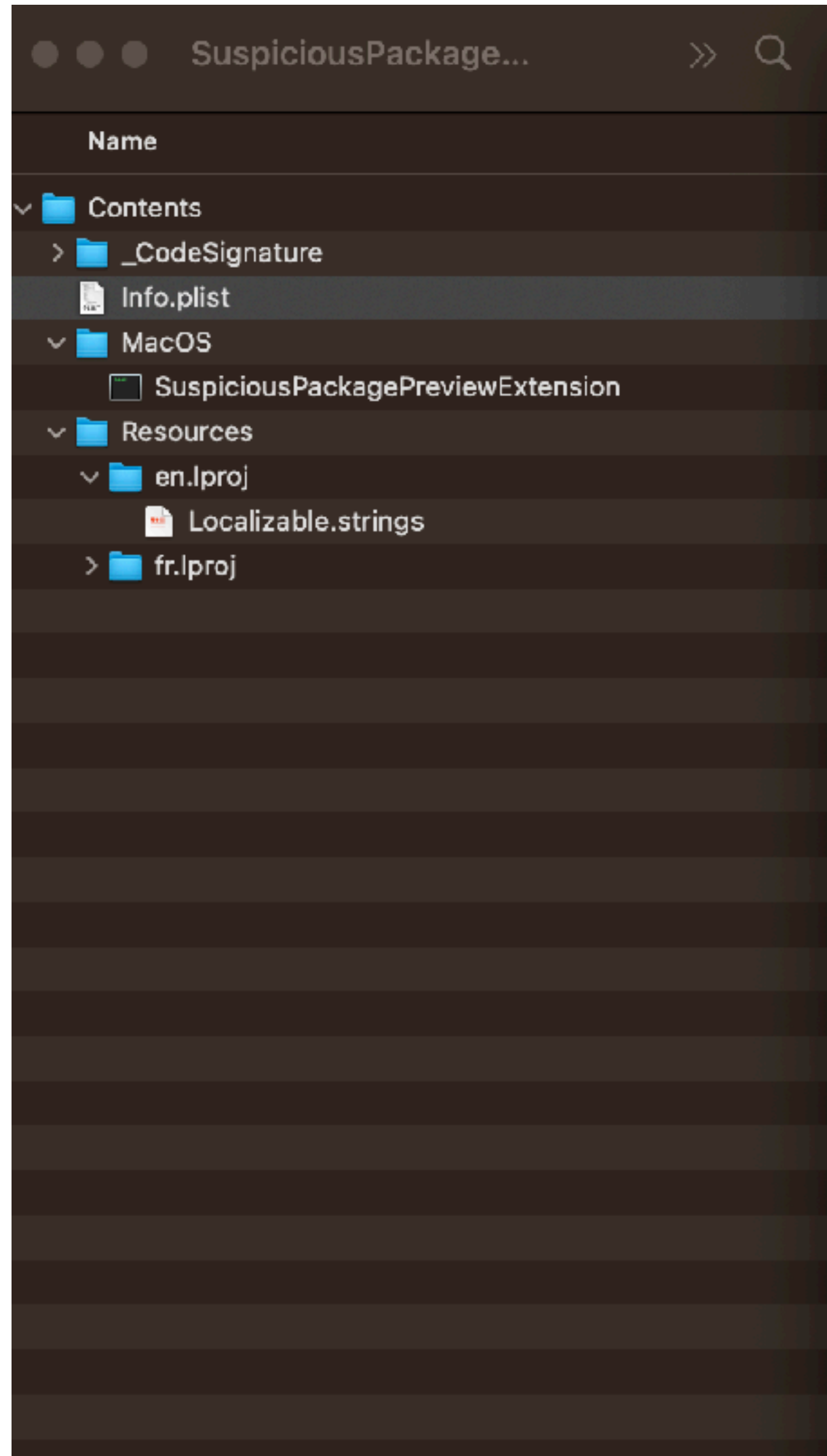


### Finder

Quick Actions, Preview







The screenshot shows the 'Info.plist' file in Xcode, displaying a table of key-value pairs. The table has three columns: Key, Type, and Value. The data is as follows:

Key	Type	Value
Information Property List	Dictionary	(23 items)
BuildMachineOSBuild	String	19H1323
Default localization	String	English
Bundle display name	String	Suspicious Package
Executable file	String	SuspiciousPackagePreviewExtension
Bundle identifier	String	com.mothersruin.SuspiciousPackageApp.QLPreview
InfoDictionary version	String	6.0
Bundle name	String	SuspiciousPackagePreviewExtension
Bundle OS Type code	String	XPC!
Bundle version string (short)	String	4.0
CFBundleSupportedPlatforms	Array	(1 item)
Item 0	String	MacOSX
Bundle version	String	826
DTCompiler	String	com.apple.compilers.llvm.clang.1_0
DTPlatformBuild	String	12D4e
DTPlatformName	String	macosx
DTPlatformVersion	String	11.1
DTSDKBuild	String	20C63
DTSDKName	String	macosx11.1
DTXcode	String	1240
DTXcodeBuild	String	12D4e
Minimum system version	String	10.14
MRSAssetCatalogDeploymentTarget	String	10.14
NSExtension	Dictionary	(3 items)
> NSExtensionAttributes	Dictionary	(2 items)
NSExtensionPointIdentifier	String	com.apple.quicklook.preview
NSExtensionPrincipalClass	String	MRSSuspiciousPackagePreviewExtensionViewController
Copyright (human-readable)	String	Copyright © 2008–2021, Randy Saldinger, dba Mothers Ruin Softw



# node-v16.13.0.pkg

DEVELOPER ID SOFTWARE • DISTRIBUTION ARCHIVE

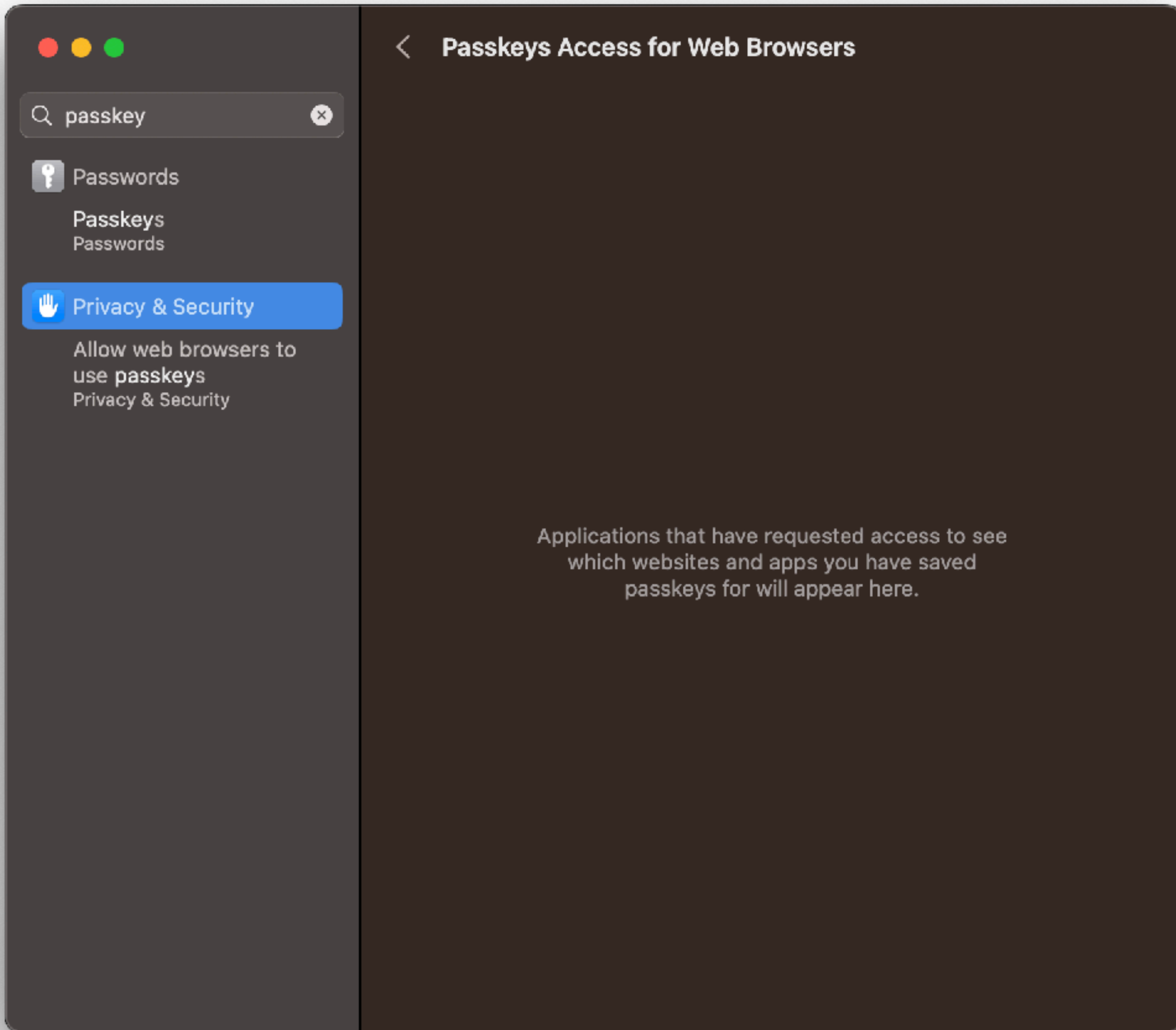
- Previously installed on Macintosh HD using macOS Installer — April 25, 20... →
- Installs 3,246 items — 168.5 MB on disk →
- DEVELOPER ID** Signed by "Developer ID Installer: Node.js Foundation (HX773...)" →
- Notarized by Apple
- Runs 2 install scripts →
- Installs natively on Apple Silicon or Intel
- Downloaded by Safari — November 24, 2021 at 2:35 AM

Open with Suspicious Package

```
charlesedge@CE-MacBook-Pro ~ % systemextensionsctl list
0 extension(s)
charlesedge@CE-MacBook-Pro ~ %
```

```
charlesedge@CE-MacBook-Pro ~ % pluginkit -mAvvv -p com.apple.authentication-services-credential-provider-ui \
| grep Path | grep -o '/.*'
/Applications/SecretChest.app/Contents/PlugIns/CredentialProvider-macOS.appex
```





< Passkeys Access for Web Browsers

passkey

Passwords

Passkeys  
Passwords

Privacy & Security

Allow web browsers to  
use passkeys  
Privacy & Security

Applications that have requested access to see  
which websites and apps you have saved  
passkeys for will appear here.



**But I just needed to make  
some of my QA tasks simpler**

**To future proof systems,  
understand how they came to be**

**The Apple II was the crowning  
achievement of the hobbyist  
movement**



# ZX Spectrum





**People explored with type-in  
programs**

A white Macintosh SE computer is shown from a three-quarter front view. It features a built-in screen, a keyboard area with a trackball, and a floppy disk drive. The text "Macintosh SE" is visible on the front panel, along with the Apple logo. The computer is set against a plain white background.

**Then came the Macintosh**

Image Courtesy of: [https://commons.wikimedia.org/wiki/File:Macintosh\\_SE\\_b.jpg](https://commons.wikimedia.org/wiki/File:Macintosh_SE_b.jpg)

**GUIs made computers more  
about utility**

**The Macintosh 512K in 1985  
Had A 20MB Hard Drive**

**Enter the kernel**





### System Folder

Name	Size	Kind	Label	Last Modified
▶ Apple Menu Items	—	folder	—	Sun, Apr 22, 192
Clipboard	zero K	file	—	Sun, Apr 22, 192
▶ Control Panels	—	folder	—	Sun, Apr 22, 192
▶ Extensions	—	folder	—	Sun, Apr 22, 192
Finder	376K	file	—	Sun, Apr 22, 192
▶ Fonts	—	folder	—	Mon, Aug 27, 195
▶ Preferences	—	folder	—	Sun, Apr 22, 192
Scrapbook File	32K	file	—	Thu, Apr 25, 1991
▶ Startup Items	—	folder	—	Sat, Apr 14, 1923
<b>System</b>	1,008K	suitcase	—	Sun, Apr 22, 1923
System Enabler 003	16K	file	—	Mon, Aug 27, 1956



Library	576K	Shared Library Ma...	—	Mon, Nov 24, 1997, 12:00 PM
Library	472K	Shared Library Ma...	—	Mon, Nov 24, 1997, 12:00 PM
8k	64K	document	—	Mon, Nov 24, 1997, 12:00 PM
	8K	system extension	—	Mon, Aug 27, 1956, 1:02 PM



System				
Name	Date Modified	Size	Kind	
> IntelligencePlatform	Mar 24, 2023 at 8:26 PM	--	Folder	
> InternetAccounts	Mar 24, 2023 at 8:26 PM	--	Folder	
> Isp	Mar 24, 2023 at 8:26 PM	--	Folder	
> Jet	Mar 24, 2023 at 8:26 PM	--	Folder	
> KerberosPlugins	Mar 24, 2023 at 8:26 PM	--	Folder	
> KernelCollections	Mar 24, 2023 at 8:26 PM	--	Folder	
▾ Kernels	Mar 24, 2023 at 8:26 PM	--	Folder	
kernel	Mar 24, 2023 at 8:26 PM	18 MB	Unix Ex...able File	
kernel.release.t6000	Mar 24, 2023 at 8:26 PM	14 MB	Document	
kernel.release.t6020	Mar 24, 2023 at 8:26 PM	14 MB	Document	
kernel.release.t8103	Mar 24, 2023 at 8:26 PM	14 MB	Document	
kernel.release.t8112	Mar 24, 2023 at 8:26 PM	13.8 MB	Document	
kernel.release.vmapple	Mar 24, 2023 at 8:26 PM	13.2 MB	Document	
> Keyboard Layouts	Mar 24, 2023 at 8:26 PM	--	Folder	
> Keychain	Mar 24, 2023 at 8:26 PM	--	Folder	
> Keychains	Mar 24, 2023 at 8:26 PM	--	Folder	
> LASecureIO	Mar 24, 2023 at 8:26 PM	--	Folder	
> LaunchAgents	Mar 24, 2023 at 8:26 PM	--	Folder	

# INITs

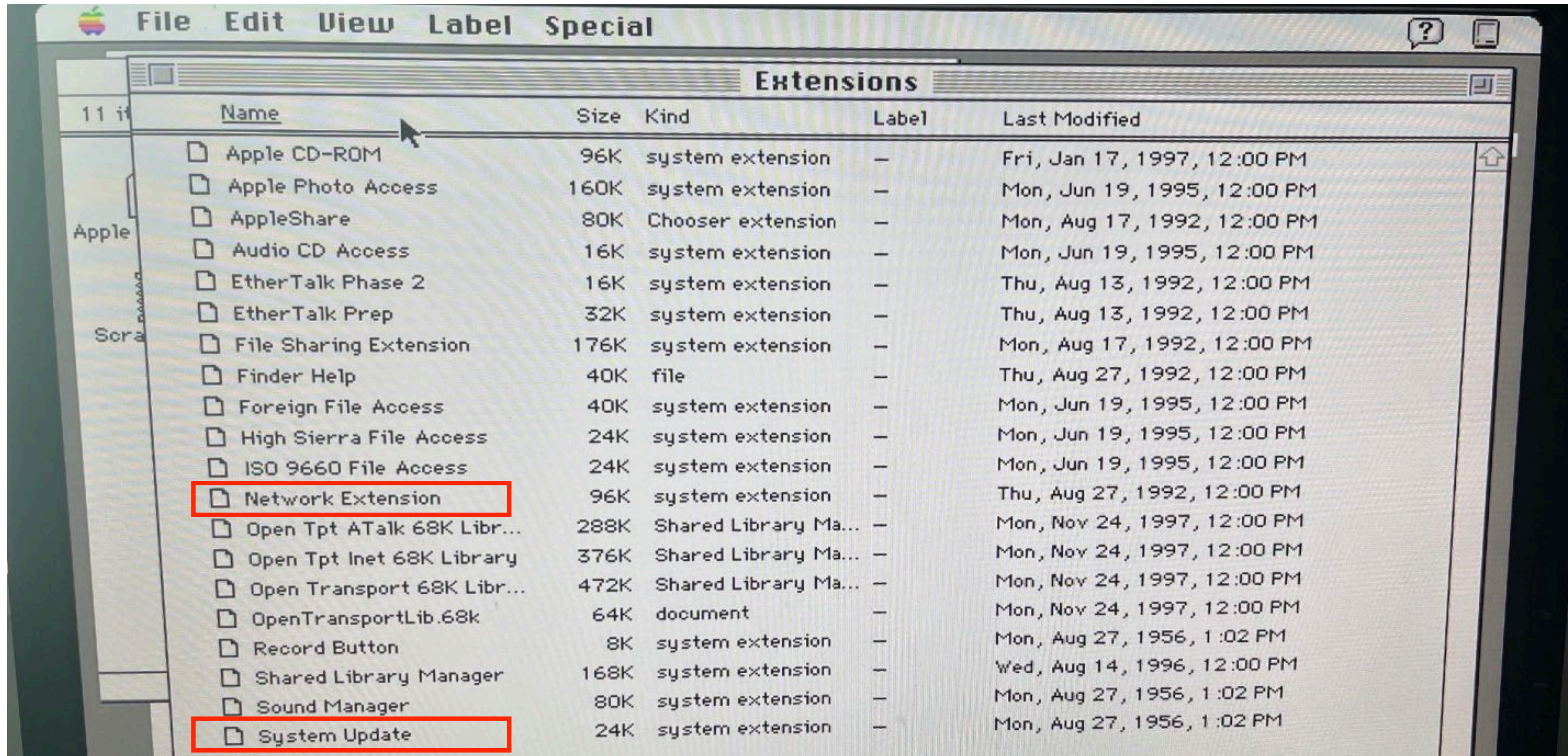
- Hackers experimented with the System suitcase and Finder file in System
- There was an INIT loader (like /etc/rc) that searched for resources at boot
- Developers figured out how to use ResEdit, to edit the resource fork of the System suitcase to INIT their own code
- Could INIT 32 files by dropping them in the System folder
- Thus we got Desk Accessories



# Desk Accessories to Extensions

- Desk Accessories were written to be drivers, but could call SystemTasks
- By System 6, suitcases stored fonts and Desk Accessories
- DA/Mover could move Desk Accessories
- Developers (and Apple) could create Control Panels as well
- Some Control Panels were hybrid extensions





# System 7

The LC III Era



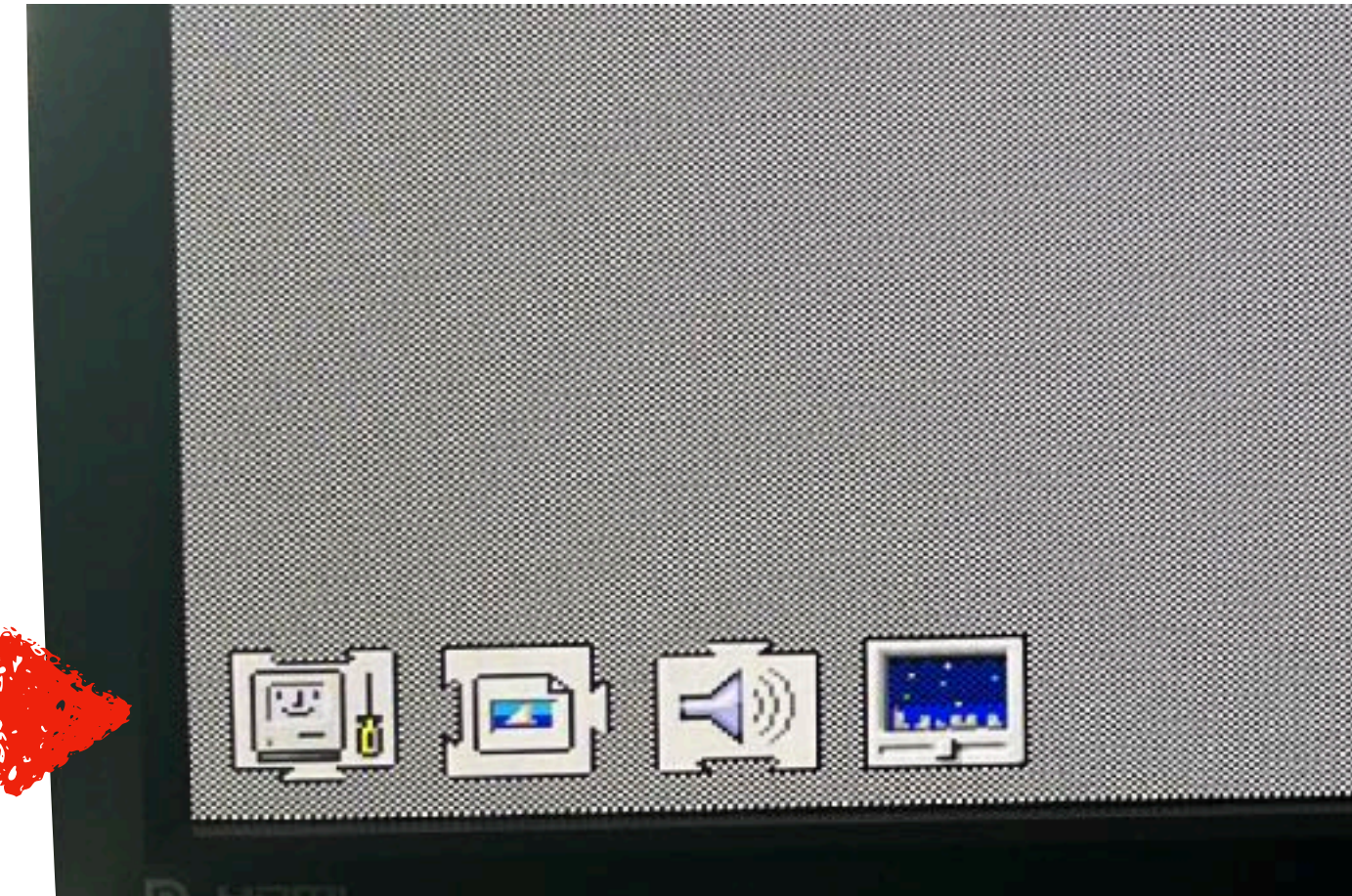
Name	Date Modified	Size	Kind
> DuetKnowledgeBase	Mar 24, 2023 at 8:26 PM	--	Folder
> ExtensionKit	Mar 24, 2023 at 8:26 PM	--	Folder
▼ Extensions	Mar 24, 2023 at 8:26 PM	--	Folder
acfs.kext	Mar 24, 2023 at 8:26 PM	3 KB	Kernel Extension
acfsctl.kext	Mar 24, 2023 at 8:26 PM	3 KB	Kernel Extension
AFKACIPCKext.kext	Mar 24, 2023 at 8:26 PM	9 KB	Kernel Extension
afps.kext	Mar 24, 2023 at 8:26 PM	5 KB	Kernel Extension
AFTK_Kext.kext	Mar 24, 2023 at 8:26 PM	3 KB	Kernel Extension
AGXFirmwareKextG13GRTBuddy.kext	Mar 24, 2023 at 8:26 PM	4 KB	Kernel Extension
AGXFirmwareKextG13XRTBuddy.kext	Mar 24, 2023 at 8:26 PM	4 KB	Kernel Extension
AGXFirmwareKextG14GRTBuddy.kext	Mar 24, 2023 at 8:26 PM	4 KB	Kernel Extension
AGXFirmwareKextG14PRTBuddy.kext	Mar 24, 2023 at 8:26 PM	4 KB	Kernel Extension
AGXFirmwareKextG14XRTBuddy.kext	Mar 24, 2023 at 8:26 PM	4 KB	Kernel Extension
AGXFirmwareKextRTBuddy64.kext	Mar 24, 2023 at 8:26 PM	4 KB	Kernel Extension
AGXG13G.kext	Mar 24, 2023 at 8:26 PM	6 KB	Kernel Extension
AGXG13X.kext	Mar 24, 2023 at 8:26 PM	6 KB	Kernel Extension
AGXG14G.kext	Mar 24, 2023 at 8:26 PM	6 KB	Kernel Extension
AGXG14X.kext	Mar 24, 2023 at 8:26 PM	5 KB	Kernel Extension
AGXGPURawCounterBundle.bundle	Mar 24, 2023 at 8:26 PM	3 KB	bundle
AGXMetal13_3.bundle	Mar 24, 2023 at 8:26 PM	5.7 MB	bundle
AGXMetalG13X.bundle	Mar 24, 2023 at 8:26 PM	10.4 MB	bundle
AGXMetalG14G.bundle	Mar 24, 2023 at 8:26 PM	5.7 MB	bundle
AGXMetalG14X.bundle	Mar 24, 2023 at 8:26 PM	7.3 MB	bundle
ALF.kext	Mar 24, 2023 at 8:26 PM	4 KB	Kernel Extension
AMD9500Controller.kext	Mar 24, 2023 at 8:26 PM	23 KB	Kernel Extension
AMD10000Controller.kext	Mar 24, 2023 at 8:26 PM	21 KB	Kernel Extension
AMDFramebuffer.kext	Mar 24, 2023 at 8:26 PM	4 KB	Kernel Extension
AMDMTLBronzeDriver.bundle	Mar 24, 2023 at 8:26 PM	108 KB	bundle
AMDRadeonServiceManager.kext	Mar 24, 2023 at 8:26 PM	5 KB	Kernel Extension
AMDRadeonVADriver.bundle	Mar 24, 2023 at 8:26 PM	2 KB	bundle
AMDRadeonVADriver2.bundle	Mar 24, 2023 at 8:26 PM	2 KB	bundle
AMDRadeonX4000.kext	Mar 24, 2023 at 8:26 PM	6 KB	Kernel Extension
AMDRadeonX4000GLDriver.bundle	Mar 24, 2023 at 8:26 PM	1.5 MB	bundle
AMDRadeonX4000HWServices.kext	Mar 24, 2023 at 8:26 PM	17 KB	Kernel Extension
AMDRadeonX5000.kext	Mar 24, 2023 at 8:26 PM	11 KB	Kernel Extension
AMDRadeonX5000GLDriver.bundle	Mar 24, 2023 at 8:26 PM	2 MB	bundle

Today  
 /System/Library/Extensions



# Less of the Wild West

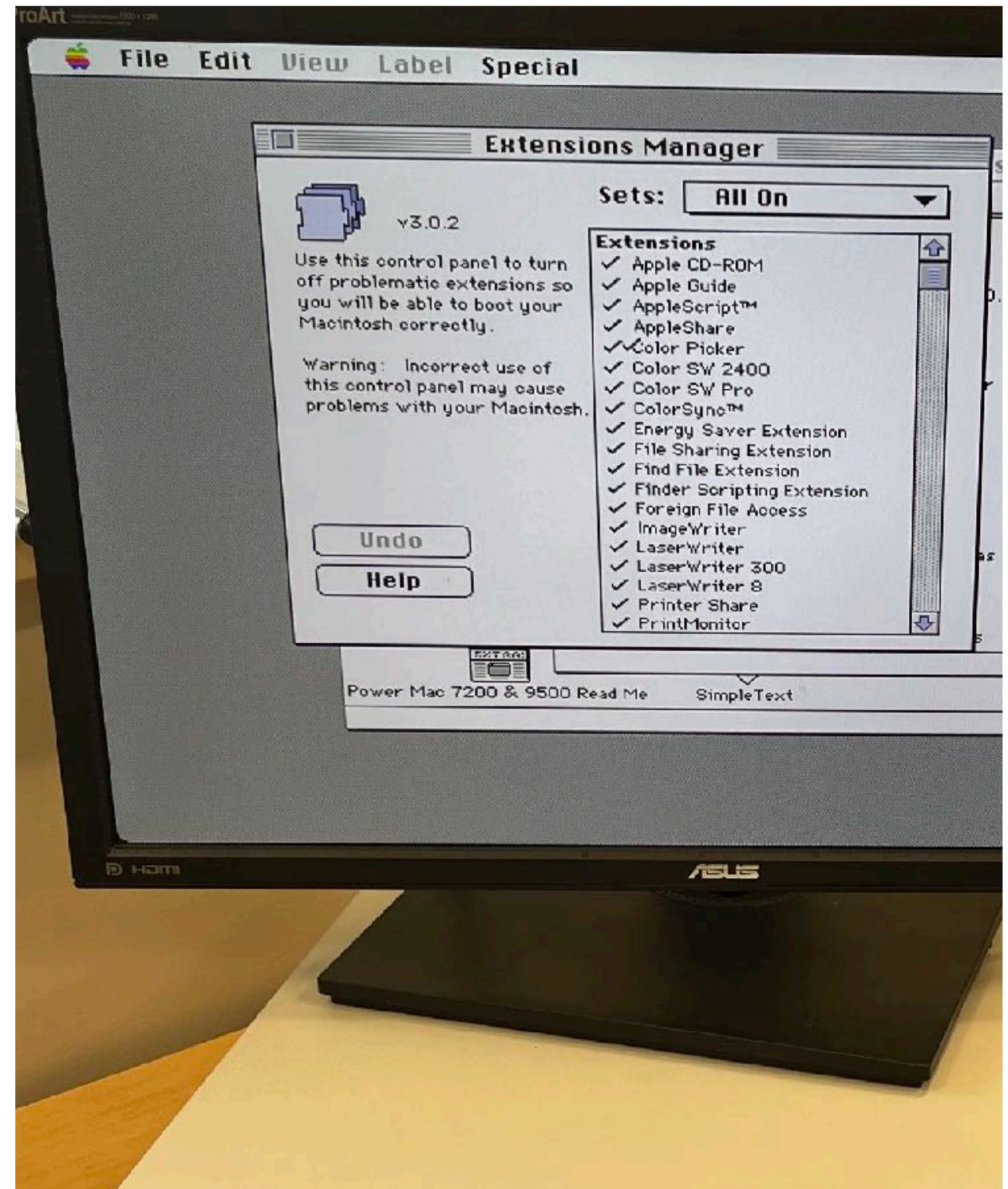
- Early extensions had a resource type of INIT as well
- The boot loader showed extension icons at boot time →
- Extensions could try to run in the same part of the system - conflicts
- Jeff Robin (who later wrote SoundJam) then wrote Conflict Catcher
- Macintosh System 7.5 added a graphical interface called Extensions Manager (a name that should sound familiar)





# Extensions Manager

Circa 1991





# Maturing Classic Extension Types

- Application Program (APPL): Programs that lived in the Extensions folder
- Chooser Extensions: Network handlers, printers, scanners, etc
- Communications Toolbox (cbnd/fbnd/tbnd): Custom socket handlers
- Document: Document handlers (e.g. Sherlock)
- Library (libr/shlb): Shared libraries and open transport
- System Extension: Described on the next slide



# Classic System Extensions

- Background Application (appe): Think of these as services or launched items now
- Component (thng): Shared code dynamically loaded when needed.
- Driver (comd for 68000 systems and ndrvc for PowerPC): Drivers
- INIT (INIT): Modifies system resources at startup and can talk directly to the kernel (similar to what later kernel extensions were).
- Scripting (scri): Never used this, just read it exists in documentation.

**The list grew to include hundreds  
from Apple and third parties**

**But conflicts...**



**Management?**



# At Ease (1992-1998)

## for using At Ease 3.0 with System 7.5

At Ease 3.0 is compatible with System 7.5. However, some control panels and other features of System 7.5 may work differently while At Ease is running. This section provides information on these differences. Please refer to Macintosh Guide, available in the Guide (📖) menu, for more information on these topics.

### AppleScript

Scripts written using AppleScript can be launched by At Ease. However, scripts that require the Finder will not run properly.

### Extensions Manager

You cannot change the settings in this control panel while At Ease 3.0 is running.

### Find File

When At Ease 3.0 is open, the Find File command does not let you open or



**Then Came OS X**

**Two Mach-based projects  
became one**



# Copland

- Developers knew the limitations the original Mac team had baked in
- Apple developed a kernel called Nukernel based on Mach
- Copland was a “failed” project that introduced protected memory that would eventually lead to multithreading
- Apps could invoke a new thread as a separate process in protected memory that could communicate with the kernel, akin to kernel extensions
- Apple released Mac OS 8 instead and bought NeXT

# The NeXT Legacy

- UNIX was initially free but AT&T then Novell got cranky about licensing after BSD extended Unix and redistributed their work with the BSD license
- The Mach micro-kernel was written at Carnegie Mellon University by Richard Rashid and Avie Tevanian
- Existing BSD 4.3 code provided compatibility with other Unix variants
- The gist: Mach + BSD + Nukernel (also Mach-based) evolved to XNU, the hybrid kernel in modern Apple operating systems



# Pipes

- With Mach came Unix Pipes; data could move between programs
- Tasks, message queues, memory management, etc gave Mach object oriented multiprocessing capabilities.
- Less monolithic but with interprocess communications (IPC), which evolved to xpc ( see [https://knight.sc/reverse\\_engineering/2020/03/20/audit-tokens-explained.html](https://knight.sc/reverse_engineering/2020/03/20/audit-tokens-explained.html) for XPC security)
- Services could be piped system wide
- NeXTSTEP got Objective-C as a programming language and what the developers called kits to provide an extensible application layer

ExtensionKit

Name	Date Modified	Size	Kind
ExtensionPoints	Mar 24, 2023 at 8:26 PM	--	Folder
com.apple.authenticat...l-provider-ui.appexpt	Mar 24, 2023 at 8:26 PM	2 KB	Document
com.apple.extensionkit.appexpt	Mar 24, 2023 at 8:26 PM	399 bytes	Document
com.apple.Safari.content-blocker.appexpt	Mar 24, 2023 at 8:26 PM	1 KB	Document
com.apple.Safari.extension.appexpt	Mar 24, 2023 at 8:26 PM	2 KB	Document
com.apple.Safari.web-extension.appexpt	Mar 24, 2023 at 8:26 PM	2 KB	Document
Extensions	Mar 24, 2023 at 8:26 PM	--	Folder
AboutExtension.appex	Mar 24, 2023 at 8:26 PM	1.6 MB	Applica...xtension
AccessibilitySettingsExtension.appex	Mar 24, 2023 at 8:26 PM	15 MB	Applica...xtension
AirDrop & Handoff.appex	Mar 24, 2023 at 8:26 PM	1.1 MB	Applica...xtension
Appearance.appex	Mar 24, 2023 at 8:26 PM	1.6 MB	Applica...xtension
AppExtensionManagement.appex	Mar 24, 2023 at 8:26 PM	172 KB	Applica...xtension
AppleIDSettings.appex	Mar 24, 2023 at 8:26 PM	2.4 MB	Applica...xtension
Bluetooth.appex	Mar 24, 2023 at 8:26 PM	1.9 MB	Applica...xtension
CDs & DVDs Settings Extension.appex	Mar 24, 2023 at 8:26 PM	470 KB	Applica...xtension
ClassKitSettings.appex	Mar 24, 2023 at 8:26 PM	1.5 MB	Applica...xtension
ClassroomSettings.appex	Mar 24, 2023 at 8:26 PM	2.4 MB	Applica...xtension
ControlCenterSettings.appex	Mar 24, 2023 at 8:26 PM	1.4 MB	Applica...xtension
DateAndTime Extension.appex	Mar 24, 2023 at 8:26 PM	1 MB	Applica...xtension
DesktopSettings.appex	Mar 24, 2023 at 8:26 PM	2.1 MB	Applica...xtension
DisplaysExt.appex	Mar 24, 2023 at 8:26 PM	5.7 MB	Applica...xtension
Extensions.appex	Mar 24, 2023 at 8:26 PM	1.1 MB	Applica...xtension
FamilySettings.appex	Mar 24, 2023 at 8:26 PM	1.4 MB	Applica...xtension
FocusSettingsExtension.appex	Mar 24, 2023 at 8:26 PM	738 KB	Applica...xtension
FollowUps.appex	Mar 24, 2023 at 8:26 PM	339 KB	Applica...xtension
GameCenterMacOSSettingsExtension.appex	Mar 24, 2023 at 8:26 PM	657 KB	Applica...xtension
GameControllerMacSettings.appex	Mar 24, 2023 at 8:26 PM	2.7 MB	Applica...xtension
HeadphoneSettingsExtension.appex	Mar 24, 2023 at 8:26 PM	1.5 MB	Applica...xtension
Internet Accounts.appex	Mar 24, 2023 at 8:26 PM	1.3 MB	Applica...xtension
KeyboardSettings.appex	Mar 24, 2023 at 8:26 PM	3.3 MB	Applica...xtension
Localization.appex	Mar 24, 2023 at 8:26 PM	1.5 MB	Applica...xtension
LockScreen.appex	Mar 24, 2023 at 8:26 PM	1 MB	Applica...xtension
LoginItems.appex	Mar 24, 2023 at 8:26 PM	550 KB	Applica...xtension
Mouse.appex	Mar 24, 2023 at 8:26 PM	23.5 MB	Applica...xtension
Network.appex	Mar 24, 2023 at 8:26 PM	8 MB	Applica...xtension
NotificationsSettings.appex	Mar 24, 2023 at 8:26 PM	1.1 MB	Applica...xtension



Favorites

- Bootstrappers Dro...
- AirDrop
- Recents
- Applications
- Downloads
- On My Mac
- Podcasterating
- Creative Cloud Files

iCloud

- iCloud Drive
- Documents
- Desktop
- Shared
- Podcasterating

Locations

- OK Computer
- Macintosh HD
- OneDrive
- Network

Tags

- Red
- Gray
- Blue
- install
- Yellow
- Orange



**More capabilities are better to  
help recruit developers**

**But kernel extensions were  
fairly complicated**

# Kernel Extensions

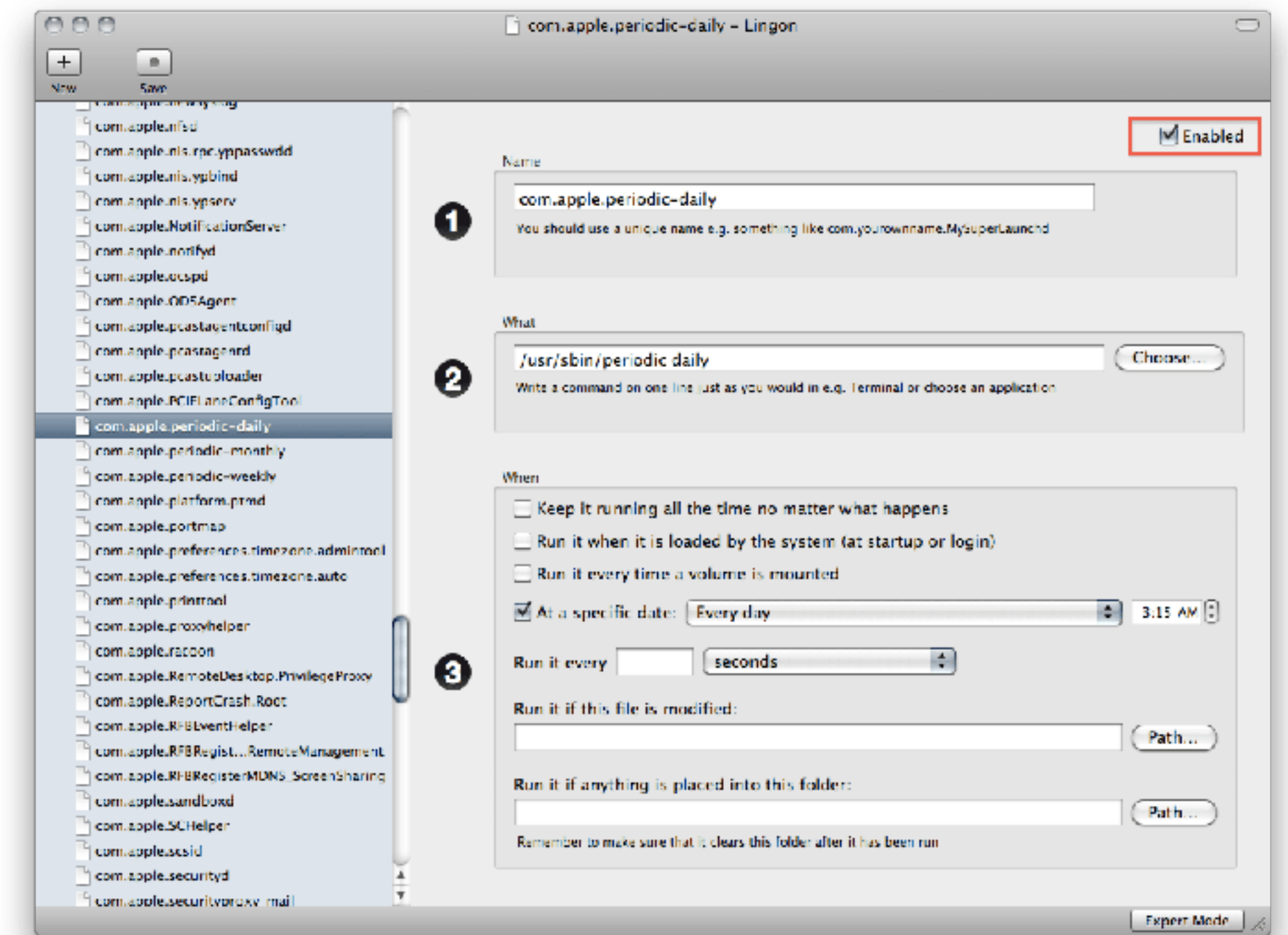
- Kernel tasks weren't officially supported
- Kernel extensions were added to expose various kernel data structures
- Users relied on kernel extensions so kernel advancements slowed
- Apple supported Kernel Programming Interfaces in 10.4 (Tiger) as part of the Kernel Authorization subsystem (kauth) <https://developer.apple.com/library/archive/technotes/tn2127/index.html>
- 10.4 brought kernel “concurrency” (no more kernel locking, so better for multi-CPU computers)
- These were mostly device, file, and network I/O tasks



**Conflict Catcher for Mac OS X  
was never released**

# Instead

- Less problems
- Command line tools like kextutil, kextload/kextunload, kextstat, kextlibs
- Peter Borg's Lingon (launchd)
- More .dylibs (third parties needed their own shared objects)



**With Unix underneath, Mac OS  
X became a hacker OS**

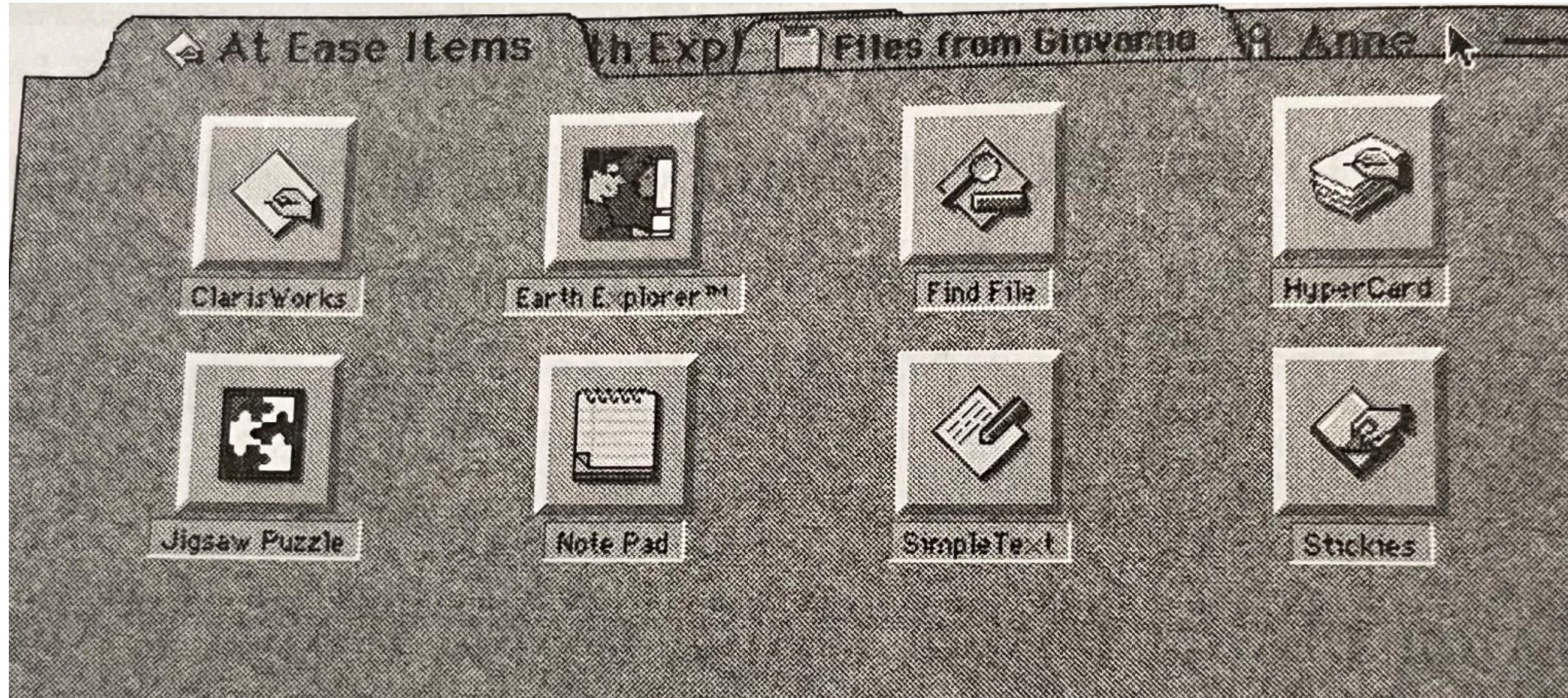


**Developer Relations didn't  
have much recourse yet...**

**Then Came iOS in 2007**



# Remember The Simple Finder?





# Modernized



**iOS didn't support kernel  
extensions**

# The Application Sandbox came to the Mac in 10.7 (2011)

Key	Type	Value
Entitlements File	Dictionary	(2 items)
App Sandbox	Boolean	NO
Access User Selected Files (Read Only)	Boolean	YES



**But its origins lie in the “TrustedBSD  
MAC Framework: Extensible Kernel  
Access Control” (2000)**

<https://ieeexplore.ieee.org/document/1194871>

**As bandwidth increased, most  
apps talked to rest endpoints, not  
through sockets**

**Cryptographic Signing became  
more pervasive  
(thank you CryptoKit)**



**MDM required a certificate**

**Soon, access to certain APIs,  
libraries, kits, etc had to be signed**

**Powerful APIs needed to be  
protected**



**iOS 2 gave us entitlements in 2008,  
which came to the Mac in 2011**

**Code signing, notarization,  
sandbox profiles, and attestation  
frameworks matured**

**Now, Apple could revoke  
certificates for bad behavior**



**The marketing term for some of  
this became Gatekeeper**

# **The importance of rootless (expanded to SIP) - 2015**

**What good is SIP if a kernel extension can trespass anywhere?**



**Example: Managed Open In  
and supervision**

# **User Accepted Kernel Extension Loading in 10.13**

**There's always backlash**





♪ Hey, what gives  
you the right ♪

**Apple forged on, with Privacy  
Preferences Policy Control**

**But the path we're on followed  
the same route it did in System 8**



**Which brings us to Part II:  
The Present**

**In the meantime...**

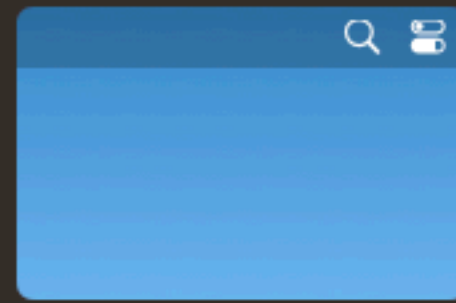
**Apple has provided us with a  
manicured sandbox**



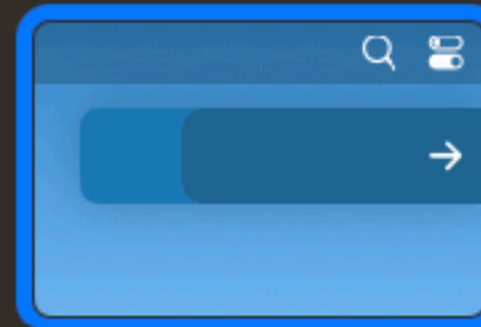
< SecretChest



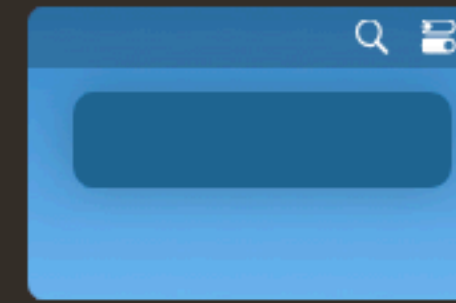
Allow notifications  
SecretChest



None



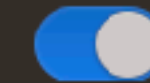
Banners



Alerts

Banners appear in the upper-right corner and go away automatically. Alerts stay on screen until dismissed.

Show notifications on lock screen



Show in Notification Center



Show previews

Default



Notification grouping

Automatic



**There are the privacy && security  
issues with what extensions can  
access**

**So it's on developers to surface the level of telemetry we want to give**



# A few extra lines of code...

```
///
extension Interface_ReadableMainActorValue<(any Interface_LocalSecretChestInstance)?> {

    ///
    public func fetchPasswordForCredentialProviderExtension
        (credentialIdentity: ASPasswordCredentialIdentity,
         extensionContext: ASCredentialProviderExtensionContext)
    -> AnyProcedure<Void> {

        ///
        Name("#function") {
            self
                .getCurrentValue
                .unwrap(orElse: "The extension has no instance to use for fetching the password.")
                .flatMap { secretChestInstance in
                    secretChestInstance
                        .loggedInSecretChest
                        .waitUntilNotNil()
                        .flatMap { loggedInInstance in
                            loggedInInstance
                                .fetchInternetPassword(
                                    for: .init(credentialIdentity)
                                )
                                .flatMap { password in
                                    loggedInInstance
                                        .disableNearbyInteraction
                                        .map { _ in extensionContext.complete(with: password) }
                                }
                            }
                        }
                    }
            }
        }
    }
}
```

Close

NearbyInstanceInteraction ✨

2

RawNearbyInstanceManager ✨

AdvertiserManager ✨

BrowserManager ✨

setupTopLevelDisplay ✓

1

waitUntilNotNil(timeout:) ✓

then ✓

()

0

setupSubscription ✓

then ✓

tryToProduceOutput(using:retryOnEachPublishFrom:timeout:) ✓

Optional<ProductionLocalSecretChestInstance> ✓
Optional<ProductionLocalSecretChestInstance> ✓

0

startupTasks ✓

Close

```

MainActorValueModule_main_actor_value.MainActorValue<Swift.Dictionary<interface_nearby_secret_chest_interaction.NearbySecretChestI
D, interface_nearby_secret_chest_interaction.Interface_NearbySecretChest>>(fetchValue: (Function)), retentionUntilDeinit:
MainActorValueModule_main_actor_value_source.MainActorValueSource<Swift.Array<Any>>, pushNotificationSender:
production_secret_chest_push_notification_sender.ProductionSecretChestPushNotificationSender(awsSNSClient: AWSSNS.SNSClient,
userDevicesManager: production_user_devices_manager.ProductionUserDevicesManager(userDevices: (Extended Existential)))))
localInstallationParticipant: installation_participant.InstallationParticipant(deviceID: C2D3E766-093A-41C1-99A4-BD4DD1382173,
instanceType: secret_chest_instance_type.SecretChestInstanceType.mainApp, secureEnclave:
local_secure_enclave.LocalSecureEnclave(_encrypt: (Function), _decrypt: (Function)), shardStorage:
local_shard_storage.LocalShardStorage(_storeShard: (Function), _fetchShard: (Function), _deleteShard: (Function), _wipeShards:
(Function)), installationMapStorage: local_installation_map_storage.LocalInstallationMapStorage(locallyStoredInstallationMaps:
MainActorValueModule_main_actor_value_source.MainActorValueSource<Swift.Dictionary<internet_password_identity.InternetPasswordIde
ntity, installation_maps.InternetPasswordInstallationMap>>, autofillSuggestionsSetter:
local_autofill_suggestions_setter.LocalAutofillSuggestionsSetter(store: local_autofill_identity_store.LocalAutofillIdentityStore(systemStore:
<ASCredentialIdentityStore: 0x6000034561d0>))), localInstallationMapStorage:
local_installation_map_storage.LocalInstallationMapStorage(locallyStoredInstallationMaps:
MainActorValueModule_main_actor_value_source.MainActorValueSource<Swift.Dictionary<internet_password_identity.InternetPasswordIde
ntity, installation_maps.InternetPasswordInstallationMap>>, autofillSuggestionsSetter:
local_autofill_suggestions_setter.LocalAutofillSuggestionsSetter(store: local_autofill_identity_store.LocalAutofillIdentityStore(systemStore:
<ASCredentialIdentityStore: 0x6000034561d0>))), localShardStorage: local_shard_storage.LocalShardStorage(_storeShard: (Function),
_fetchShard: (Function), _deleteShard: (Function), _wipeShards: (Function)), _locallyStoredInstallationMaps:
MainActorValueModule_main_actor_value_source.MainActorValueSource<Swift.Dictionary<internet_password_identity.InternetPasswordIde

```





DASHBOARD

PROVIDER CONFIG

USER

SECRETS

SSO CONFIG

PAYMENT

LOGS

GROUPS

PINCODE

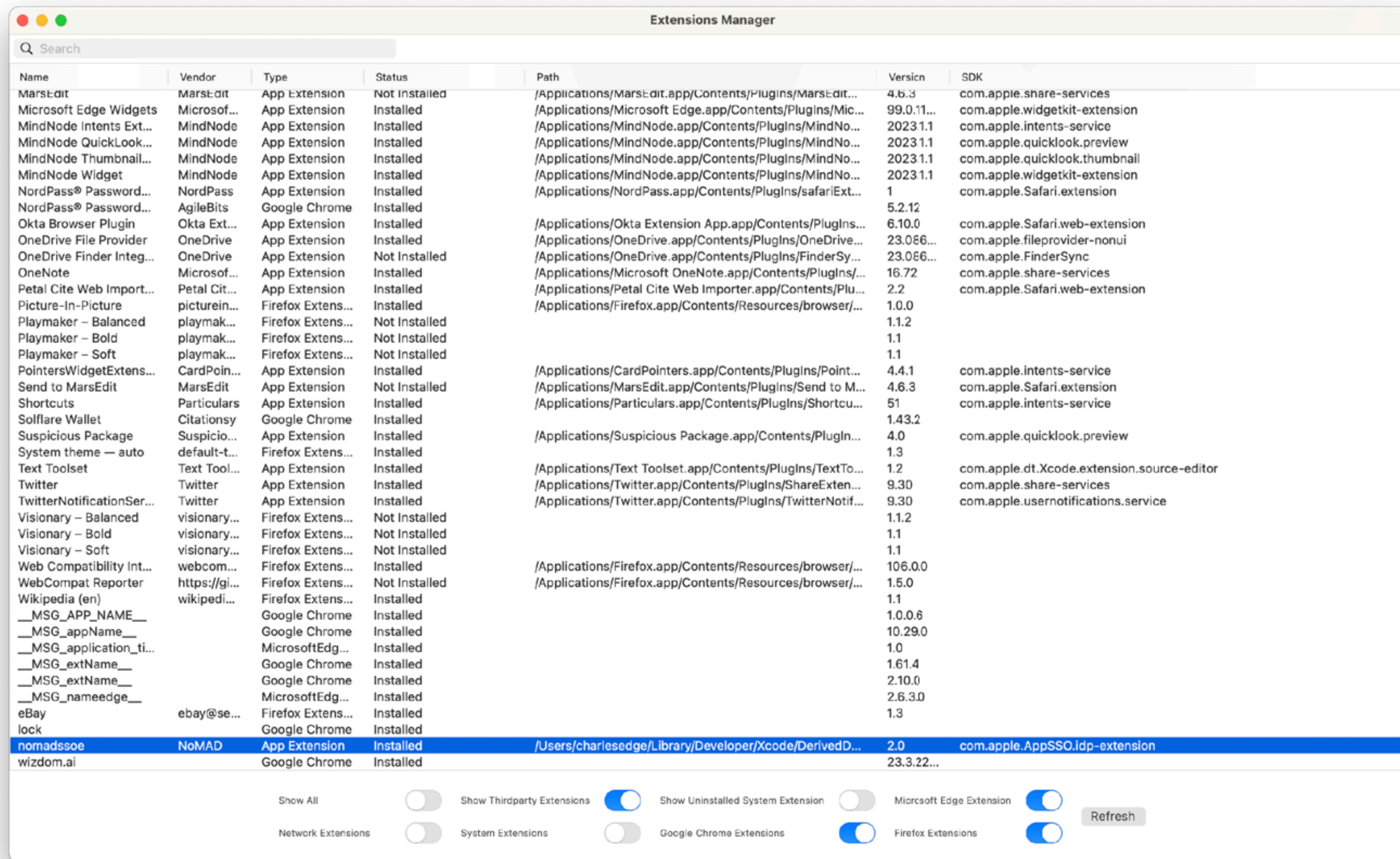
## Logs

Module Name	Updated Time	API	Status
UsersDeviceList	05/16/2023 : 12:49PM	GET /auth/users/device/list	↑ Success
UsersDeviceList	05/16/2023 : 12:49PM	GET /auth/users/device/list	↑ Success
UsersDeviceList	05/16/2023 : 12:49PM	GET /auth/users/device/list	↑ Success
UsersDeviceList	05/16/2023 : 12:49PM	GET /auth/users/device/list	↑ Success
UsersDeviceList	05/16/2023 : 12:49PM	GET /auth/users/device/list	↑ Success

« Previous **1** 2 3 4 5 ... 10 Next »

PINCODE

« Previous **1** 2 3 4 5 ... 10 Next »



<https://github.com/krypted/extensionsmanager>



**Thank You!**