

# Es tut sich wieder was bei Broadcom WLAN

Beitrag von „DerBeste“ vom 30. November 2025, 19:57

## [Zitat von apfelnico](#)

Hab zwar keine Ahnung welches Problem gelöst wurde, habe aber große Hochachtung vor der Programmierung einer signierten DEXT. Also modern wie es sein soll. Von so etwas habe ich in unserem Umfeld kein zweites Mal gehört. Meinen Glückwunsch!

Hi Apfelnico,

Die folgenden Ausführungen basieren auf umfangreichen Eigenentwicklungen und praktischen Erfahrungen aus mehrwöchiger Arbeit mit macOS-Treibern. Angesichts der spärlichen und oft unzureichenden offiziellen Veröffentlichungen seitens Apple musste ich viele der hier dargestellten Erkenntnisse durch ausgiebiges Experimentieren und intensives Testen gewinnen.

Apple liefert nach meinem Kenntnisstand nur minimale Dokumentation zu den neuen DEXT-Technologien und verschweigt bewusst wichtige Details, was die Entwicklung erheblich erschwert.

Hier eine Gesamtübersicht: KEXT vs. DEXT und die Zukunft von Hackintosh

Die Entwicklung von Treibern unter macOS befindet sich in einem tiefgreifenden Wandel. Apple verfolgt die klare Strategie, die traditionellen Kernel Extensions (KEXT) vollständig abzulösen und durch moderne Driver Extensions (DEXT) zu ersetzen. Dieser Wandel hat erhebliche Auswirkungen auf alle Entwickler, insbesondere auf Community- und Hobby-Programmierer.

## 1. Der technische Wandel: KEXT zu DEXT

### 1.1 Kernel Extensions (KEXT) - Die alte Architektur

KEXTs sind Erweiterungen, die direkt im Kernel von macOS ausgeführt werden. Diese Architektur bietet zwar volle Kontrolle und schnellen Hardwarezugriff, bringt aber erhebliche Nachteile mit sich:

- Direkter Zugriff auf Kernel-Ressourcen
- Hohe Anfälligkeit für Systemabstürze (Kernel Panics)
- Manuelle Nutzerfreigabe seit macOS High Sierra
- Erhöhte Systemanfälligkeit durch fehlerhafte Module

KEXTs gelten heute offiziell als veraltet und werden von Apple langfristig nicht mehr unterstützt.

## 1.2 Driver Extensions (DEXT) - Die moderne Lösung

Driver Extensions sind Teil des System Extension Frameworks und laufen vollständig im User Space. Sie sind sicherer, stabiler und moderner:

- Keine Kernel-Ausführung → deutlich weniger Systemabstürze
- Abgesicherte, sandbox-basierte Umgebung
- Müssen in ein App-Bundle integriert werden
- Nutzung moderner, restriktiver APIs
- Abhängigkeit vom System Extension Management

DEXTs sind zwar sicherer, aber erheblich komplizierter zu entwickeln und bieten weniger Zugriffsrechte als klassische KEXTs.

## 2. Apples strategischer Kurs seit 2017

Eigentlich ist seit macOS High Sierra klar erkennbar, dass Apple die KEXTs schrittweise aus dem System entfernt. Die wichtigsten Meilensteine:

- 2017 - High Sierra: Einführung von User-Approved Kernel Extension Loading
- 2019 - Catalina: Start des System Extension Frameworks
- 2020 - Big Sur: Signed System Volume und Klassifizierung von KEXTs als "Legacy"
- 2021-2024: Weitere Einschränkungen und Abschaffung zahlreicher Kernel-APIs

Diese Entwicklung zeigt: Apple will KEXTs vollständig eliminieren.

## 3. Auswirkungen auf Entwickler und Community-Projekte

Die Umstellung auf DEXTs bringt erhebliche Einschränkungen mit sich:

- Keine direkten Kernelhooks mehr
- Stark limitierte Zugriffsmöglichkeiten auf Hardware
- Treiber müssen in Apps verpackt werden
- Nur eingeschränkt dokumentierte APIs

- Höherer Aufwand beim Debugging
- Zunehmende Abhängigkeit von signierten Entitlements

Damit werden viele etablierte Vorgehensweisen der Hackintosh-Community deutlich erschwert.

#### 4. Das nahende Ende der Hackintosh-Ära

Mit der erwarteten finalen macOS-Generation macOS 26.x, die voraussichtlich die letzte Version mit x86\_64-Unterstützung sein wird, steht der Hackintosh-Community ein tiefgreifender Einschnitt bevor.

Die zentralen Punkte:

- Apple stellt seine gesamte Produktlinie auf Apple Silicon um
- Ab macOS 26.x ist sehr wahrscheinlich Schluss mit neuen x86\_64-Builds
- Ohne x86\_64-Kernel existiert keine Basis mehr für Hackintosh-Installationen
- Aktuelle und zukünftige Treiberentwicklungen (DEXT) sind zunehmend auf Apple Silicon ausgelegt
- Der klassische PC-basierte Hackintosh wird mittelfristig nicht mehr realisierbar sein

Damit ist klar: Wir bewegen uns auf das tatsächliche Ende des Hackintosh-Ökosystems zu.

#### 5. Spezielle Problematik: PCI Netzwerkkarten

Die bisherigen Netzwerk- und PCI-Treiber wie die beliebten Realtek RTL8xxx Karten werden nicht mehr von den modernen DEXT-DriverKit unterstützt. Man kann zwar versuchen, einen Kext dafür zu bauen, aber mit vielen Hürden.

Alternative zu IONetworkController / IOPCIDevice für Realtek-Ethernetkarten

In modernen macOS-Versionen ist die klassische KEXT-Architektur weitgehend abgelöst worden. Besonders Netzwerk- und PCI-Treiber betroffen:

Klassisch (KEXT)	DriverKit (Modern)
IONetworkController	NetworkDriverKit
IOPCIDevice	IOPCI (DriverKit PCI Family)
Kernel-Level Netzwerktreiber	Userspace-basierter Netzwerktreiber
Voller Hardwarezugriff	Starke Sicherheits-Einschränkungen

Die moderne API IOPCI erlaubt grundlegende Geräte-Erkennung, Lesen/Schreiben in PCI-Konfiguration und Mapping von BAR-Memory, aber nicht denselben Tiefgang wie IOPCIDevice im Kernel.

Für Netzwerkkarten hat Apple das Framework NetworkDriverKit eingeführt, das IONetworkController ersetzt. Damit erhält man eine moderne Userspace-taugliche API, Unterstützung für Netzwerk-Interfaces und Queues für RX/TX.

Fazit für Realtek-Chips:

Ein vollständiger RTL8168/8111-Treiber wie früher lässt sich in DriverKit nicht mehr 1:1 umsetzen. Eine realistische Lösung besteht aus PCI-Handling in einer minimalen Kernel-Extension und Ethernet-Interface in einer DriverKit SystemExtension.

## 6. WLAN-Adapter unter macOS Tahoe

### 6.1 USB-WLAN-Adapter - höchste Erfolgchance

USB-Adapter sind aktuell die einzige realistische und dauerhaft tragfähige Basis für WLAN-Treiber unter macOS Tahoe, da sie über das User-Space-fähige IOUSBHost-Framework laufen.

Gute Chancen bestehen für:

- Realtek RTL8812AU / BU (802.11ac)
- Realtek RTL8822BU (802.11ac)
- Mediatek MT7612U (802.11ac)
- Mediatek MT7921U (WiFi 6)

### 6.2 PCIe-WLAN-Karten - möglich, aber extrem schwierig

PCIe-Treiber über DEXT sind theoretisch machbar, aber extrem schwierig, weil:

- DMA, Interrupts und Registerzugriffe stark eingeschränkt sind
- Apple den gesamten WLAN-Stack vollständig blockiert
- AWDL, Handoff etc. nicht nachbildbar sind
- DEXT kein direkter Ersatz für die alten Broadcom-KEXTs ist

Chancen nach Chipsätzen:

- BCM4360: niedrig-mittel (noch am ehesten machbar)
- BCM43602: niedrig (komplexere Firmware, wenig Dokumentation)
- BCM4352: niedrig (bekannte Limitierungen)

### 6.3 Apple-eigene WLAN-Module - praktisch ausgeschlossen

Alle Chipsätze, die Apple in Macs verbaut, sind:

- tief in das AppleIO80211-Framework integriert
- stark signiert
- vollständig proprietär
- ohne DEXT-Äquivalent
- ohne veröffentlichte APIs

Dazu gehören BCM4350, BCM4364, BCM4377, BCM4387 und alle Apple Silicon internen WiFi/BT-Module.

### 7. Fazit

- KEXTs waren lange das Fundament vieler Community-Treiber, verlieren aber endgültig ihre Bedeutung
- DEXTs sind die Zukunft, allerdings restriktiver, komplexer und weniger flexibel
- Apples klare Sicherheitsstrategie erschwert freie Treiberentwicklung erheblich
- Mit macOS 26.x nähert sich die Hackintosh-Ära dem Abschluss, da keine neuen x86-Builds mehr zu erwarten sind
- Für Netzwerkkarten und WLAN-Adapter bedeutet dies: nur noch sehr eingeschränkte Möglichkeiten, vor allem über USB-basierte Lösungen

Die Entwicklergemeinschaft steht vor der Herausforderung, sich an diese neuen Gegebenheiten anzupassen oder alternative Lösungen zu finden.

Problemstellung mit dem RTL802.11ac.dext Treiber unter macOS Tahoe:

Leider verliefen die Testergebnisse mit meinem RTL802.11ac.dext unter macOS Tahoe nicht erfolgreich. Während der DEXT-Treiber unter macOS Sequoia problemlos mittels des Installations-Tools installiert wird und korrekt erkannt wird, sieht die Situation unter macOS Tahoe derzeit sehr kritisch aus.

Zwar lässt sich die Installation mit dem Tool ohne Fehlermeldungen durchführen, jedoch kommt es anschließend zu keiner funktionsfähigen Initialisierung des Treibers.

Trotz ordnungsgemäßer Vorgehensweise:

Korrekter Aufbau und Einbindung des DEXT-Pakets

Vollständige Code-Signierung nach Apple-Richtlinien

Fehlerfreie Installation durch das zertifizierte Tool

Ich muss die Ursache für dieses Verhalten unbedingt identifiziert und beheben.

### **Edit:**

So ein Ärger. Ich habe oben ausführlich erklärt, dass man für den Build das passende KDK, also das Kernel Development Kit, installieren muss. Und genau das habe ich selbst völlig übersehen. Mir ist erst jetzt wieder eingefallen, dass ich es noch nicht eingerichtet habe. Das

muss ich unbedingt nachholen.

