

BETA-TESTPROGRAMM · MAI 2026

Amiga Imager

Beta-Testhandbuch v0.78

VERTRAULICH — NUR FÜR TESTER

Vielen Dank, dass du dir die Zeit nimmst, Amiga Imager zu testen. Dieses Dokument führt dich durch Installation, Bedienung und die wichtigsten Testszenarien.

PiStorm / Emu68

Classic Amiga

MiSTer FPGA

Version 0.78 Beta

Plattform macOS 14 Sonoma oder neuer

Datum Mai 2026

Status Geschlossener Beta-Test

Inhalt

1. Überblick & Systemvoraussetzungen

2. Installation

- 2.1 App installieren
- 2.2 Erster Start
- 2.3 Benötigte Dateien vorbereiten

3. Die Benutzeroberfläche

- 3.1 Einfacher Modus vs. Erweiterter Modus
- 3.2 Karte 1 — Eingabe (Input)
- 3.3 Karte 2 — Konfiguration (Configuration)
- 3.4 Karte 3 — Software
- 3.5 Karte 4 — Image & Build
- 3.6 Einstellungen (Cmd+,)

4. Plattform PiStorm / Emu68

- 4.1 Voraussetzungen
- 4.2 Schritt-für-Schritt-Anleitung
- 4.3 FrameThrower (USB-Kamera)

5. Plattform Classic Amiga

- 5.1 Voraussetzungen
- 5.2 Hardware-Auswahl
- 5.3 Schritt-für-Schritt-Anleitung

6. Plattform MiSTer FPGA

- 6.1 Voraussetzungen
- 6.2 Schritt-für-Schritt-Anleitung

7. Testprotokoll

- 7.1 Allgemeine Tests (alle Plattformen)
- 7.2 PiStorm-spezifische Tests
- 7.3 Classic-spezifische Tests
- 7.4 Software-Pakete testen

8. Fehlerbehebung (Troubleshooting)

9. Feedback geben

1 Überblick & Systemvoraussetzungen

Amiga Imager ist eine native macOS-Anwendung, die bootfähige Amiga-Disk-Images für drei Plattformen erstellt: PiStorm/Emu68, Classic Amiga und MiSTer FPGA. Die App übernimmt alle Schritte vollautomatisch — von der Partitionierung bis zur Software-Installation.

Anforderung	Minimum	Empfohlen
macOS	14.0 Sonoma	15 Sequoia oder neuer
Prozessor	Apple Silicon oder Intel	Apple Silicon (M1 oder neuer)
Speicher	4 GB RAM	8 GB RAM
Festplatte	500 MB freier Platz	2 GB (für Downloads)
Internetverbindung	Erforderlich	Breitband (Aminet-Downloads)
SD-Karte / CF-Karte	8 GB	32 GB oder mehr

⚠ Achtung: Die App benötigt **AmigaOS-Installationsmedien** (ADF-Dateien) und eine **Kickstart-ROM**. Diese Dateien sind urheberrechtlich geschützt und werden nicht mitgeliefert. Du benötigst eine legale Lizenz (z.B. Cloanto Amiga Forever oder originale Datenträger).

2 Installation

2.1 App installieren

- 1 Die mitgelieferte `.zip`-Datei entpacken (Doppelklick genügt).
- 2 **Amiga Imager.app** in den *Programme*-Ordner ziehen.
- 3 Beim ersten Start erscheint möglicherweise eine Sicherheitswarnung. Über *Systemeinstellungen* → *Datenschutz & Sicherheit* → *Dennoch öffnen* bestätigen. Die App ist signiert und notariert.

2.2 Erster Start

Beim ersten Start lädt die App automatisch benötigte Hilfstools herunter:

- **hst-imager** — Partitionierungswerkzeug für Amiga RDB
- **hst-amiga** — Werkzeug für Amiga-Icon-Bearbeitung
- **lha** — Archivierungsprogramm für Aminet-Pakete

Dieser Vorgang dauert ca. 1–2 Minuten. Der Fortschritt wird in der App angezeigt.

Hinweis: Wenn die automatische Installation fehlschlägt, auf `Install / Retry` in der App klicken.

2.3 Benötigte Dateien vorbereiten

Folgende Dateien werden für den Build benötigt. Am besten alle in einem gemeinsamen Ordner ablegen:

Datei	Beschreibung	Dateiname (Beispiel)
AmigaOS Workbench ADF	Installations-Disketten des Amiga-Betriebssystems	<code>Workbench3.2.adf</code>
Kickstart ROM	Kickstart-Datei des Amiga (je nach Modell)	<code>kick.rom</code> / <code>Kickstart3.2.rom</code>
Modules ADF <i>(optional)</i>	Für KS 3.1 → OS 3.2 Upgrade auf Classic-Hardware	<code>Modules.adf</code>
P96 iComp-Archiv <i>(optional)</i>	Picasso96 RTG-Treiber für Classic RTG-Karten	<code>Picasso96.lha</code>

3 Die Benutzeroberfläche

3.1 Einfacher Modus vs. Erweiterter Modus

Die App hat zwei Modi, umschaltbar über den Schalter oben rechts:

Modus	Karten	Geeignet für
Einfach	Input, Basis-Konfiguration, Software, Image & Build	Schnelle Builds mit Standardeinstellungen
Erweitert	Input, vollständige Konfiguration, Software, Image & Build	Alle Einstellungen, individuelle Hardware

✓ **Tipp:** Für den Beta-Test bitte immer den **Erweiterten Modus** verwenden, um alle Funktionen zu testen.

3.2 Karte 1 — Eingabe (Input)



Zielplattform

Segmentierter Schalter oben mit drei Optionen:

- **PIStorm / Emu68** — Raspberry Pi als Beschleuniger im Amiga
- **Classic Amiga** — Echte Amiga-Hardware (A500 bis A4000)
- **MiSTer FPGA** — MiSTer mit Minimig-Core

AmigaOS-Version

Automatische Erkennung aus den ADF-Dateinamen. Mit **Check Media** kann die Erkennung manuell ausgelöst werden. Falls die Erkennung falsch ist, kann die Version manuell erzwungen werden.

Kickstart ROM

Kickstart-ROM-Datei auswählen. Die App prüft automatisch die Kompatibilität mit der gewählten Plattform. Für PiStorm wird das ROM in die FAT-Boot-Partition geschrieben.

Workbench ADF-Dateien

Installationsmedien für AmigaOS auswählen. Mehrere ADFs können gleichzeitig ausgewählt werden (Workbench + Extras + Fonts + Modules etc.).

Lokale

Optionale Lokalisierungs-ADF (Deutsch, Französisch usw.).

Ausgabe-Image

Zieldatei für das fertige Disk-Image. Standardmäßig auf dem Desktop.

Classic Amiga — Hardware-Auswahl

Nur sichtbar bei Plattform :

Feld	Beschreibung	Beispielwerte
Machine	Amiga-Modell	A500, A600, A1200, A2000, A3000, A4000, A4000T
Accelerator	Beschleunigerkarte (CPU)	None, 68030, 68040, 68060
RTG Graphics Card	RTG-Grafikkarte	None, Picasso II, Picasso IV, CyberVision 64, ZZ9000, ...
Network Card	Netzwerkkarte (SANA-II)	None, Ariadne, X-Surf 100, PCMCIA 3Com, ...

3.3 Karte 2 — Konfiguration (Configuration)

① Input

② **Configuration**

③ Software

④ Image & Build

Diese Karte ist in vier Abschnitte unterteilt:

Display

Einstellung	Beschreibung
Screen mode	RTG-Auflösung (z.B. 1280×720, 1920×1080). Für Classic ohne RTG: native Modi (PAL/NTSC/HIRES)
WB Color depth	Farbtiefe des Workbench-Bildschirms (4, 8, 16 oder 32 Bit)
Backdrop	Hintergrundbild einschalten/ausschalten
Icon set	GlowIcons oder 4ColorIcons (beeinflusst das Aussehen aller Icons)
P96 iComp archive	Optionales Picasso96-Archiv von iComp (für Classic RTG-Karten)

i Hinweis: PiStorm verwendet immer RTG (VideoCore). Die Auflösung wird direkt in die `screenmode.prefs` geschrieben und beim ersten Boot aktiviert.

FrameThrower (nur PiStorm)

Einstellungen für die angeschlossene USB-Kamera (FrameThrower-Modul):

Einstellung	Beschreibung
Enable FrameThrower	Kamera-Unterstützung aktivieren
Mode	Smooth (bilinear, weicher) oder Integer (pixelgenau)
Autostart	Kamera startet automatisch nach dem Boot
Scaling, Offset, Brightness, Contrast	Kamera-Bildeinstellungen

Network

Einstellung	Beschreibung
Network stack	None — kein Netzwerk; Roadshow — TCP/IP-Stack (Freeware-Version enthalten, eigene Lizenz ebenfalls nutzbar)
WiFi SSID / Password	WLAN-Zugangsdaten für PiStorm (WirelessManager)

Transfer Folder

Optionaler Ordner auf dem Mac, der beim Build auf eine separate Partition des Images kopiert wird (praktisch für eigene Dateien und Spiele).

3.4 Karte 3 — Software

① Input

② Configuration

③ **Software**

④ Image & Build

Optionale Software-Pakete aus Aminet, in Kategorien gruppiert. Alle mit einem Häkchen markierten Pakete werden automatisch heruntergeladen und installiert.

Kategorie	Pakete (Auswahl)
System	MUI 3.8, StackAttack, icon.library 46
Multimedia	AHI (Audiotreiber), IMP3 (MP3-Player), AmigaAMP, RiVA
Utilities	iGame (Spiele-Launcher), TankMouse / HID2AMI (USB-Maus)
Network	LumiFTP (FTP-Client), IBrowse (Webbrowser), WHDLoad
Appearance	WBDock (Dock für Workbench), GlowIcons / 4ColorIcons

i Hinweis: Einige Pakete haben versteckte Abhängigkeiten (z.B. wird beim Aktivieren von AmigaAMP automatisch *asyncio.library* und *mpega.library* mitinstalliert). Die Auswahl kümmert sich selbst darum.

3.5 Karte 4 — Image & Build

① Input

② Configuration

③ Software

④ **Image & Build**

Image-Größe

Voreinstellungen: 8 GB, 32 GB, 64 GB, 128 GB, 256 GB oder benutzerdefiniert. Die App zieht automatisch einen Puffer ab, damit das Image auf eine echte SD-Karte passt (konfigurierbar in Einstellungen).

Build starten

- Build Image** — Erstellt das Image (Datei auf der Festplatte)
- Build & Write to SD** — Erstellt das Image und schreibt es direkt auf die ausgewählte SD-Karte

Ein Fortschrittsbalken zeigt den aktuellen Schritt an (Partitionierung → ADF-Extraktion → Asset-Overlay → Software-Download → Abschluss). Die geschätzte Restzeit wird angezeigt.

⚠ Achtung: Den Build-Prozess nicht unterbrechen. Ein halbfertiges Image ist nicht bootfähig. Bei Problemen kann der Build einfach erneut gestartet werden.

3.6 Einstellungen (Cmd+,)

Seltener benötigte Optionen im macOS-Einstellungsfenster:

Einstellung	Beschreibung
Debug mode	Detaillierte Log-Ausgabe (für Fehlerberichte)
Use UserFiles	Eigene Dateien aus <code>~/Library/Application Support/Amiga-Imager/Assets/UserFiles/</code> automatisch erkennen
Floppy drive	DF0: im Image einrichten
SD Card Size Reduction	Puffer pro Kartengröße (Standard: 300–1500 MB)
PiStorm: Use Prereleases	Emu68-Beta-Versionen verwenden (wichtig für PiStorm16)
PiStorm: Boot Mode	Emu68-Boot-Modus (Auto, FAT, RDB)

4 Plattform: PiStorm / Emu68

4.1 Voraussetzungen

- Amiga mit eingebautem PiStorm-Modul (PiStorm32Lite, PiStorm16, oder klassischer PiStorm)
- Raspberry Pi 3 oder 4 / CM4 auf dem PiStorm
- SD-Karte (empfohlen: 32 GB oder mehr, A1/A2-Klasse)
- AmigaOS 3.2 Workbench + Extras + Fonts ADF-Dateien
- Kickstart-ROM (kompatibel mit dem Amiga-Modell)

i Hinweis: Emu68 ist eine vollständige 68040-Emulation — das heißt, es läuft auf jedem A500/A600/A1200 ohne physischen Beschleuniger, als wäre ein echter 68040 verbaut.

4.2 Schritt-für-Schritt-Anleitung

- 1 Plattform wählen:** Karte 1 →
- 2 Medien auswählen:** ADF-Dateien, Kickstart-ROM und Ausgabedatei angeben.
- 3 Bildschirmmodus:** Karte 2 → Display → wählen (empfohlen: 1280×720 oder passend zum Monitor).
- 4 Netzwerk:** wählen. Für WLAN SSID und Passwort eintragen.
- 5 Software:** Karte 3 → gewünschte Pakete auswählen (Empfehlung: MUI, AHI, WBDock, iGame, WHDLoad).
- 6 Image-Größe:** Karte 4 → Kartengröße wählen (32 GB für den Start empfohlen).
- 7 Build starten:** → SD-Karte aus der Liste wählen. Build dauert ca. 10–20 Minuten.
- 8 Fertig:** SD-Karte auswerfen, in den PiStorm einsetzen und Amiga einschalten.

i Hinweis: Beim ersten Boot läuft automatisch eine Einrichtungsroutine (OneTimeRun). Der Amiga startet dabei einmal neu — das ist normal.

4.3 FrameThrower (USB-Kamera)

Der FrameThrower erlaubt es, das Bild einer USB-Kamera direkt auf dem Amiga-RTG-Bildschirm anzuzeigen — ein Videodigitizer für die moderne Ära.

- 1 Karte 2 → **FrameThrower** → **Enable FrameThrower** einschalten.
- 2 Modus wählen: **Smooth** für weiche Skalierung oder **Integer** für pixelgenaue Darstellung.
- 3 USB-Kamera an den Raspberry Pi anschließen, dann Build starten und auf SD-Karte schreiben.
- 4 Nach dem Boot: Kamera startet automatisch, wenn **Autostart** aktiviert ist.

5 Plattform: Classic Amiga

5.1 Voraussetzungen

- Einen echten Amiga (A500, A600, A1200, A2000, A3000, A4000 oder A4000T)
- CF-Karte oder SD-Karte mit Adapter (IDE → CF oder PCMCIA → SD je nach Modell)
- AmigaOS 3.x ADF-Dateien und passendes Kickstart-ROM
- Für RTG: Kompatible P96-Grafikkarte und optionaler iComp-Archiv

⚠ **Achtung:** Classic-Amiga-Images werden ohne FAT-Boot-Partition erstellt. Das Kickstart kommt aus dem ROM-Sockel, nicht aus dem Image.

5.2 Hardware-Auswahl

Maschine

Das Amiga-Modell bestimmt die verfügbaren Optionen. Zum Beispiel:

- **A1200** — eingebauter 68020, PCMCIA-Slot für NIC
- **A4000T** — eingebauter Zorro-III-Bus, unterstützt ZZ9000 und andere High-End-Karten
- **A500/A600** — Stock 68000, kein RTG ohne Erweiterung

Beschleuniger

Option	Bedeutung
None	Stock-CPU des Amiga-Modells
68030	Beschleunigerkarte mit 68030 (z.B. ACA630, Blizzard 1230)
68040	Beschleunigerkarte mit 68040 (z.B. Blizzard 1240)
68060	Beschleunigerkarte mit 68060 (z.B. Blizzard 1260, TF1260)

RTG-Grafikkarte

Unterstützte Karten: Picasso II/II+/IV, CyberVision 64/64-3D, Retina BLT, Spectrum, Piccolo, Piccolo SD64, A2410, Domino, Merlin, UAEGFX, ZZ9000.

📌 **Hinweis:** Für jede RTG-Karte wird der passende Picasso96-Treiber automatisch konfiguriert.

Netzwerkkarte

Unterstützte SANA-II-Treiber: Ariadne, Ariadne II, A2065, Hydra, X-Surf, X-Surf 100, EB920, LanRover, PCMCIA NE2000, PCMCIA 3Com.

5.3 Schritt-für-Schritt-Anleitung

- 1 **Plattform:** Karte 1 →
- 2 **Hardware:** Maschine, Beschleuniger, RTG-Karte und Netzwerkkarte entsprechend der eigenen Hardware wählen.
- 3 **Medien:** Workbench-ADF (+ ggf. Modules-ADF für KS 3.1), Kickstart-ROM und ggf. P96-iComp-Archiv angeben.
- 4 **Bildschirmmodus:** RTG-Modus (wenn Grafikkarte gewählt) oder nativer Modus (PAL/NTSC).
- 5 **Netzwerk:** wählen (wenn Netzwerkkarte vorhanden).
- 6 **Build und Schreiben** wie bei PiStorm — CF-/SD-Karte in den Amiga einbauen und starten.

6 Plattform: MiSTer FPGA

6.1 Voraussetzungen

- MiSTer FPGA mit installiertem Minimig-Core
- SD-Karte für MiSTer (separate vom MiSTer-System)
- AmigaOS 3.x ADF-Dateien

i Hinweis: MiSTer-Images sind reine HDF-Dateien (kein FAT-Boot). Das Kickstart stellt der Minimig-Core bereit.

6.2 Schritt-für-Schritt-Anleitung

- 1 **Plattform:** wählen.
- 2 **RTG (optional):** Karte 2 → *Enable UAEGFX RTG* aktivieren (erfordert 68020 im MiSTer-OSD).

- 3 **Netzwerk (optional):** Roadshow wählen. MiSTer nutzt PPP über seriell (UART). Im MiSTer-OSD: UART Mode = PPP, Baud = 115200.
- 4 Build starten und HDF-Datei auf die MiSTer-SD-Karte kopieren.

7 Testprotokoll

Bitte die folgenden Tests durchführen und die Ergebnisse notieren. Beim Feedback bitte das ausgefüllte Protokoll mitsenden (Foto oder Abschrift).

⚠ Achtung: Vor jedem Test: Debug-Modus in Einstellungen aktivieren (`Cmd+,` → Debug mode). Das ermöglicht detaillierte Logs für die Fehleranalyse.

7.1 Allgemeine Tests (alle Plattformen)

#	Test	Erwartetes Ergebnis	Bestanden?	Anmerkung
A1	App startet und lädt Tools	Kein Fehler, hst-imager / Iha verfügbar	<input type="checkbox"/>	
A2	ADF-Erkennung: Workbench-ADF auswählen	AmigaOS-Version wird korrekt erkannt	<input type="checkbox"/>	
A3	Build startet und zeigt Fortschritt	Fortschrittsbalken bewegt sich, kein Absturz	<input type="checkbox"/>	
A4	Build schlägt fehl: fehlende ROM-Datei	Verständliche Fehlermeldung, kein Absturz	<input type="checkbox"/>	
A5	Build abgeschlossen: Image-Datei vorhanden	Image-Datei mit korrekter Größe erstellt	<input type="checkbox"/>	
A6	Software-Pakete: MUI + WBDock aktivieren	Pakete werden heruntergeladen und installiert	<input type="checkbox"/>	
A7	Erster Boot des Images	Amiga startet, Workbench erscheint	<input type="checkbox"/>	
A8	Icon-Darstellung: GlowIcons oder 4ColorIcons	Farbige Icons auf Workbench sichtbar	<input type="checkbox"/>	

7.2 PiStorm-spezifische Tests

#	Test	Erwartetes Ergebnis	Bestanden?	Anmerkung
P1	Boot in RTG-Modus (1280×720)	Workbench in RTG-Auflösung, kein schwarzer Bildschirm	<input type="checkbox"/>	
P2	Anderer Bildschirmmodus (z.B. 1920×1080)	Korrekte Auflösung nach erstem Boot	<input type="checkbox"/>	
P3	HDDToolBox öffnen	Zeigt korrekt <code>brcm-emmc.device</code> (Pi4/CM4) oder <code>brcm-sdhc.device</code> (Pi3)	<input type="checkbox"/>	
P4	Netzwerk: genet/Ethernet testen	IP-Adresse erhalten, Ping funktioniert	<input type="checkbox"/>	
P5	Netzwerk: WLAN (WirelessManager)	Verbindung mit eingegebenem SSID/Passwort	<input type="checkbox"/>	
P6	Hintergrundbild (Backdrop)	Auflösungspassendes Hintergrundbild auf Workbench	<input type="checkbox"/>	
P7	WHDLload starten	WHDLload v20.0 ohne Nag-Screen	<input type="checkbox"/>	
P8	iGame öffnen	iGame startet, WBDock-Eintrag vorhanden	<input type="checkbox"/>	
P9	AmigaAMP: MP3-Datei abspielen	Audio-Wiedergabe funktioniert	<input type="checkbox"/>	
P10	IBrowse: Browser öffnen	IBrowse startet, Lesezeichen vorhanden	<input type="checkbox"/>	
P11	FrameThrower: Kamera aktivieren	Kamera-Bild erscheint auf dem Bildschirm	<input type="checkbox"/>	
P12	Prerelease-Build: Emu68 v1.1.x	Einstellungen → PiStorm: Use Prereleases aktivieren, Build starten, Log prüfen	<input type="checkbox"/>	

7.3 Classic-Amiga-spezifische Tests

#	Test	Erwartetes Ergebnis	Bestanden?	Anmerkung
C1	Boot ohne Beschleuniger (Stock-CPU)	Amiga bootet in Workbench	<input type="checkbox"/>	
C2	Boot mit Beschleuniger (68030/040/060)	ShowConfig zeigt korrekten CPU-Typ	<input type="checkbox"/>	
C3	KS 3.1 → OS 3.2 ROMUPDATE	Auf KS 3.1 Hardware: OS 3.2 lädt nach ROM-Update	<input type="checkbox"/>	
C4	RTG-Grafikkarte: P96Prefs öffnen	Korrekte RTG-Karte mit verfügbaren Modi	<input type="checkbox"/>	
C5	RTG-Modus starten	Workbench wechselt in RTG-Auflösung	<input type="checkbox"/>	
C6	Netzwerkkarte: Roadshow testen	IP-Adresse via DHCP erhalten	<input type="checkbox"/>	
C7	PCMCIA NIC (A1200): cnetdevice/3c589	Karte erkannt, DHCP erfolgreich	<input type="checkbox"/>	
C8	HDDToolBox öffnen	Zeigt korrekt <code>scsi.device</code>	<input type="checkbox"/>	

7.4 Software-Pakete testen

Paket	Was testen?	Bestanden?
MUI 3.8	MUI-basierte Apps (iGame, LumiFTP etc.) starten fehlerfrei	<input type="checkbox"/>
AHI	AHI Prefs öffnen, Soundausgabe testen	<input type="checkbox"/>
WBDock	Dock erscheint auf Workbench mit Icons	<input type="checkbox"/>
iGame	iGame öffnen, WHDLoad-Game starten	<input type="checkbox"/>
WHDLoad	Game per WHDLoad starten, kein Nag-Screen	<input type="checkbox"/>
IBrowse	Webseite laden (mit Netzwerk), Lesezeichen prüfen	<input type="checkbox"/>
LumiFTP	Verbindung zu einem FTP-Server testen	<input type="checkbox"/>
AmigaAMP	MP3-Datei abspielen	<input type="checkbox"/>
RiVA	Video abspielen (nur 68040+)	<input type="checkbox"/>
StackAttack	Im SYS:C-Ordner vorhanden, kein Absturz beim Boot	<input type="checkbox"/>

8 Fehlerbehebung (Troubleshooting)

8.1 Build-Probleme

✗ "hst-imager not found"

- In der App auf `Install / Retry` klicken.
- Internetverbindung prüfen.
- Einstellungen → `Show Folder` → prüfen ob `hst-imager` im Tools-Ordner liegt.

✗ Build bricht mit Fehlermeldung ab

- **Debug-Modus aktivieren** (Cmd+, → Debug mode) und Build wiederholen.
- Log-Datei aus `~/Library/Logs/Amiga-Imager/` für den Fehlerbericht bereitstellen.
- Häufige Ursachen: falsche ADF-Dateien, fehlende ROM-Datei, SD-Karte zu klein.

✗ "No space left on device" beim Schreiben

- Einstellungen → SD Card Size Reduction → Wert für die verwendete Kartengröße erhöhen (z.B. von 1000 auf 1500 MB).
- Oder eine kleinere Image-Größe wählen (z.B. 32 GB statt 64 GB).

✗ ADF-Erkennung schlägt fehl

- AmigaOS-Version manuell aus dem Dropdown wählen.
- ADF-Dateinamen sollten die Version enthalten (z.B. `Workbench3.2.adf`).

8.2 Boot-Probleme

✗ Amiga bootet nicht (schwarzer Bildschirm)

- **PiStorm:** SD-Karte korrekt eingesteckt? Emu68-Log auf der FAT-Partition prüfen (`emu68.Log`).
- **Classic:** ROM im richtigen Sockel? CF/SD-Karte korrekt partitioniert?
- Anderen Bildschirmmodus wählen und neu bauen (z.B. 1280×720 statt 1920×1080).

✗ Amiga bootet, aber kein RTG / falscher Bildschirmmodus

- Prüfen ob der Monitor die gewählte Auflösung unterstützt.

- Niedrigere Auflösung (800×600 oder 1024×768) probieren.
- **PiStorm:** Monitor-Datei `DEVS:Monitors/Videocore` vorhanden? Neu bauen falls nicht.
- In P96Prefs prüfen, ob die Grafikkarte korrekt erkannt wird.

✘ Guru Meditation beim ersten Boot

- **Fehlernummer 8700 xxxx** — bei Classic mit KS 3.1: Modules-ADF fehlt oder ist falsch. Modules-ADF in Karte 1 hinzufügen und neu bauen.
- Anderen Browser / Startup-Fehler: Build-Log prüfen und Feedback senden.

✘ Workbench startet, aber Icons fehlen oder sind falsch

- GlowIcons/4ColorIcons aktiviert? Karte 2 → Display → Icon set prüfen.
- Neu bauen mit explizit gewähltem Icon-Set.

8.3 Netzwerk-Probleme

✘ Roadshow startet nicht / keine IP-Adresse

- Kabel-Verbindung prüfen (LAN-Kabel an genet / Ethernet-Adapter).
- **Classic A600:** PCMCIA-Karte korrekt eingesteckt?
- **PiStorm:** genet-Treiber geladen? Im Amiga-Shell: `AddNetInterface genet` manuell versuchen.
- DHCP-Server im Netzwerk vorhanden? Mit statischer IP testen falls nicht.

✘ WLAN verbindet nicht (PiStorm)

- SSID und Passwort erneut prüfen (Groß-/Kleinschreibung beachten).
- 2,4-GHz-Netzwerk verwenden (Pi 3/4 unterstützt kein 5 GHz).
- WLAN-Netzwerk mit WPA2 konfiguriert? WPA3 wird nicht unterstützt.

8.4 Software-Pakete

✘ MUI-Fehler beim Start einer Anwendung

- MUI 3.8 in Karte 3 → Software aktivieren und neu bauen.
- Prüfen ob die MCC-Bibliotheken installiert sind (iGame benötigt sie).

✘ AmigaAMP startet, aber kein Sound

- AHI-Paket in Software-Auswahl aktiviert?
- AHI Prefs öffnen und Soundausgabe konfigurieren.

- Auf PiStorm: AudioMode auf PAULA oder AHI setzen.

✘ WHDLoad zeigt Nag-Screen

- Das sollte nicht passieren (v20.0 benötigt keine Registrierung mehr).
- WHDLoad-Version prüfen: `VERSION WHDLoad` im Shell. Muss v20.0 sein.
- Bitte als Bug melden!

✘ IBrowse öffnet sich, aber keine Lesezeichen

- Beim ersten Start legt IBrowse die Standardlesezeichen an.
- Falls keine Lesezeichen erscheinen: Bitte als Bug mit Screenshot melden.

✘ HDToolBox zeigt falsches Gerät (`scsi.device` auf PiStorm Pi4)

- Dieses Problem wurde in v0.78 behoben. Falls es noch auftritt, bitte als Bug melden.
- Korrekt sollte sein: Pi4/CM4 → `brcm-emmc.device` , Pi3 → `brcm-sdhc.device` .

9 Feedback geben

Dein Feedback ist sehr wertvoll! Bitte bei Problemen folgende Informationen mitschicken:

Was immer mitschicken

- **macOS-Version** (Apple-Menü → Über diesen Mac)
- **Amiga-Modell und Konfiguration** (z.B. A1200 + TF1260)
- **Plattform und Einstellungen** (Screenshot der App-Konfiguration)
- **Was passiert ist** (genaue Beschreibung des Fehlers)
- **Was erwartet wurde**

Log-Datei mitschicken (sehr wichtig!)

Die Log-Datei enthält alle Details des Build-Prozesses:

```
~/Library/Logs/Amiga-Imager/amigaimager-DATUM_UHRZEIT.Log
```

In Finder: *Gehe zu* → *Gehe zum Ordner ...* → `~/Library/Logs/Amiga-Imager/` → neueste Datei auswählen.

✓ **Tipp:** Debug-Modus aktivieren (Cmd+, → Debug mode) *bevor* der fehlerhafte Build gestartet wird — das erzeugt ein viel ausführlicheres Log.

Feedback-Checkliste

- Testprotokoll ausgefüllt (Seite 8–9 dieses Dokuments)
- Log-Datei aus `~/Library/Logs/Amiga-Imager/` angehängt
- macOS-Version angegeben
- Amiga-Konfiguration beschrieben
- Screenshots von Fehlermeldungen oder unerwartetem Verhalten
- Bei Boot-Problemen: Foto vom Bildschirm des Amiga