

Claude Code Debug Prompt V3: HEIC - Final Analysis & Solutions Needed

Key Discovery

iOS automatically converts HEIC → JPEG when using the photo picker or drag-and-drop. So iPhone users uploading directly from their device have NO issues.

The problem is ONLY: Desktop users who transferred HEIC files from iPhone to PC, then try to upload the raw `.heic` files.

What Failed (Console Evidence)

Client-side heic2any 0.0.4:

```
Could not parse HEIF file
{code: n, subcode: n}
[HEIC] Client-side conversion failed, trying server-side:
{code: 2, message: 'ERR_LIBHEIF format not supported'}
```

Root cause: heic2any 0.0.4 uses an old libheif WASM that doesn't support newer iPhone HEIC variants (likely HEIF with H.265 codec from iPhone 11+).

Server-side sharp:

```
[HEIC] Server-side conversion failed: 500
{"error":{"Conversion failed","details":{"source: bad seek to 1733561...
heif: Error while loading plugin: Support for this compression format has not been
built in (11.6003)}}
```

Root cause: The production environment's sharp/libvips doesn't have the libheif plugin compiled in. This is expected for cloud environments that don't include proprietary codecs.

Current State

- **heic2any:** Version 0.0.4 (this IS the latest - no newer version exists)
 - **sharp:** Version 0.34.5 installed but HEIC support requires libheif plugin not available in production
 - **iOS uploads:** Work perfectly (OS does conversion)
 - **Desktop HEIC uploads:** Fail completely
-

Options to Consider

Option 1: Use a Different Client-Side Library

Libraries to investigate:

- `libheif-js` - More actively maintained libheif WASM binding
- `heic-decode` + `jpeg-js` - Decode HEIC and re-encode as JPEG
- `@pdfng/heic` - Alternative HEIC decoder

Question for you: Which of these (if any) would work better than `heic2any` for modern iPhone HEIC files?

Option 2: Accept Limitation + User Guidance

Since iOS handles conversion automatically:

1. Show a clear warning when HEIC upload fails
2. Guide desktop users to convert HEIC to JPEG before uploading (suggest iPhone settings change or online converter)
3. Don't block the upload flow - just show informative error

Option 3: Cloud Conversion Service

Use an external API that handles HEIC conversion:

- CloudConvert API
- Imgix
- Cloudinary

Downside: Requires API key, has costs, adds external dependency

Option 4: Different Server-Side Approach

- Use `imagemagick` instead of `sharp` (but probably also lacks HEIC in production)
- Serverless function with custom `libheif` build (complex)

What I Need From You

1. **Recommendation:** Which option is most practical given:
 - iOS users already work fine (majority of users)
 - Only affects desktop HEIC uploads (minority case)
 - Production environment has limited native binary support
 - Simplicity is preferred
 2. **If Option 1:** Which library should replace `heic2any`? Provide exact code.
 3. **If Option 2:** What's the best UX for gracefully handling the failure?
 4. **Code changes needed:** Whatever you recommend, provide the exact file changes for DeepAgent to implement.
-

Files That Would Need Changes

- `/components/ui/photo-uploader.tsx` - HEIC detection and conversion logic

- `/app/api/convert-heic/route.ts` - Server-side fallback (maybe remove if not useful)
 - `package.json` - Library changes
-

Context for Decision

The app is for **vacation rental property owners** to upload photos for AI enhancement. Primary users:

- Property owners with iPhones taking photos (iOS handles conversion ✓)
- Property managers uploading from desktop (might have HEIC from transferred iPhone photos)

Given this context, what's your recommended approach?