

# Netlify Deployment Fix - ETXTBSY Error Resolution

---

## Problem

---

The CoachFlow deployment was failing on Netlify with an `ETXTBSY` (text file busy) error during the `npm install` phase. This error occurred specifically with the esbuild binary in the netlify-cli dependency.

## Error Details

```
Error: ETXTBSY (text file busy)
Location: /opt/build/repo/node_modules/netlify-cli/node_modules/esbuild/bin/esbuild
```

This is a common caching issue on Netlify where cached `node_modules` from previous builds conflict with new installations.

## Root Cause

---

1. **Cached Node Modules:** Netlify caches `node_modules` between builds to speed up deployments
2. **Binary Conflicts:** When npm tries to reinstall or update binary files (like esbuild), they're locked by the cache
3. **Inconsistent State:** The cache can get into an inconsistent state, especially with dependencies that have native binaries

## Solution Implemented

---

### 1. Updated `netlify.toml`

#### Key Changes:

- **Cache Clearing:** Added `rm -rf node_modules .cache` before npm install
- **Clean Install:** Switched from `npm install` to `npm ci` for reproducible builds
- **Node Version:** Pinned to Node 18.17.0 for consistency
- **NPM Flags:** Added `--prefer-offline --no-audit` to avoid unnecessary network calls
- **Build Processing:** Disabled Netlify's asset optimization to prevent conflicts

#### Before:

```
[build]
  command = "echo 'No build step required'"
  publish = "."
```

#### After:

```
[build]
  command = "rm -rf node_modules .cache && npm ci --prefer-offline --no-audit && echo 'Build complete'"
  publish = "."

[build.environment]
  NODE_VERSION = "18.17.0"
  NPM_FLAGS = "--prefer-offline --no-audit"
  NO_UPDATE_NOTIFIER = "true"

[build.processing]
  skip_processing = false
  [build.processing.css]
    bundle = false
    minify = false
  [build.processing.js]
    bundle = false
    minify = false
```

## 2. Created `.npmrc`

Added npm configuration to optimize behavior and prevent caching issues:

```
prefer-offline=true
audit=false
update-notifier=false
loglevel=error
progress=false
engine-strict=true
```

## 3. Generated `package-lock.json`

- Ensures consistent dependency versions across all builds
- Required for `npm ci` command
- Prevents unexpected version changes that could cause conflicts

## 4. Updated `.gitignore`

Added comprehensive ignore rules:

- `node_modules/` - Never commit dependencies
- Log files
- OS-specific files
- IDE configurations

# Why This Works

## `npm ci` vs `npm install`

- `npm ci` : Clean install from `package-lock.json` , removes `node_modules` first
- `npm install` : Updates existing `node_modules` , can cause conflicts
- Result: Fresh install every time, no stale cache issues

## Cache Clearing

- `rm -rf node_modules .cache` : Removes any cached dependencies before install
- Ensures clean slate for each build

- Prevents ETXTBSY errors from locked files

## Node Version Pinning

- Explicitly sets Node 18.17.0
- Prevents version drift between local and production
- Ensures consistent npm behavior

## Build Processing Disabled

- Netlify's asset optimization can conflict with custom build processes
- Disabling prevents double-processing of assets
- Vanilla JS doesn't need bundling/minification during build

## Testing the Fix

---

### Netlify Deploy Preview

1. Push changes to your repository
2. Netlify will automatically trigger a new build
3. Monitor the build logs for:
  - "rm -rf node\_modules .cache" (cache clearing)
  - "npm ci" (clean install)
  - "Build complete" (success message)

### Expected Build Output

```

🔧 Building site
Command: rm -rf node_modules .cache && npm ci --prefer-offline --no-audit && echo 'Build complete'

✓ Cleared cache
✓ Installing dependencies with npm ci
✓ Build complete

✓ Build succeeded

```

## Alternative Solutions (If Issues Persist)

---

### Option 1: Disable Dependency Caching

Add to `netlify.toml` :

```

[build.environment]
  NPM_CONFIG_CACHE = "/dev/null"

```

### Option 2: Use Yarn Instead

Update `package.json` :

```

"packageManager": "yarn@1.22.19"

```

And `netlify.toml` :

```
[build]
command = "rm -rf node_modules .cache && yarn install --frozen-lockfile && echo 'Build complete'"
```

## Option 3: Clear Netlify Build Cache

In Netlify UI:

1. Go to Site Settings → Build & Deploy
2. Scroll to “Build settings”
3. Click “Clear cache and retry deploy”

## Verification Checklist

- [x] `netlify.toml` updated with cache-clearing build command
- [x] `.npmrc` created with npm optimization settings
- [x] `package-lock.json` generated for consistent installs
- [x] `.gitignore` updated to exclude `node_modules`
- [x] Node version pinned to 18.17.0
- [x] Build command uses `npm ci` instead of `npm install`
- [x] Build processing disabled to prevent conflicts

## Files Modified

1. **netlify.toml** - Build configuration with ETXTBSY fix
2. **.npmrc** - NPM behavior optimization
3. **package-lock.json** - Dependency lock file (generated)
4. **.gitignore** - Updated ignore patterns

## Next Steps

1. **Push Changes:** Commit and push all changes to trigger a new build
2. **Monitor Build:** Watch Netlify build logs for successful deployment
3. **Test Site:** Verify all functionality works after deployment
4. **Environment Variables:** Ensure all required env vars are set in Netlify UI

## Required Environment Variables

Make sure these are set in Netlify UI (Site Settings → Environment Variables):

### Required

- `SUPABASE_URL`
- `SUPABASE_ANON_KEY`
- `SUPABASE_SERVICE_KEY`
- `STRIPE_SECRET_KEY`
- `STRIPE_PUBLISHABLE_KEY`
- `STRIPE_WEBHOOK_SECRET`
- `RESEND_API_KEY`

## Optional (for Stripe pricing)

- `STRIPE_PRICE_BASIC`
- `STRIPE_PRICE_PREMIUM`
- `STRIPE_PRICE_ELITE`

## Support

---

If you continue to experience ETXTBSY errors:

1. Clear Netlify build cache (Site Settings → Clear cache)
  2. Check Node version compatibility (must be  $\geq 18.0.0$ )
  3. Verify `package-lock.json` is committed to repository
  4. Try the alternative solutions listed above
- 

**Last Updated:** November 1, 2025

**Issue:** ETXTBSY error on Netlify deployment

**Status:**  Fixed