



## Issue: `_commonjsHelpers` Bundle Name and Chrome's Underscore Restriction

**Why this happens:** When Vite/Rollup processes CommonJS dependencies (like `@ricky0123/vad-web` or `onnxruntime-web`), it generates helper chunks with names like `_commonjsHelpers.<hash>.js`. Unfortunately, Chrome extensions cannot include files or directories whose names start with an underscore – Chrome treats those as system-reserved and refuses to load the extension <sup>1</sup>. This is why your unpacked MV3 extension fails to load with the error about `_commonjsHelpers...` being a reserved name.

**Failed approaches:** Direct hacks (custom `sanitizeFileName` functions, renaming scripts, post-build fixes, etc.) get overridden by WXT/Vite's build process. WXT's defaults produce a valid *published* extension, but for an unpacked extension in dev, you hit Chrome's underscore restriction. We need a configuration-level solution.

## Solution: Customize Vite/Rollup Chunk Naming in WXT

You can configure WXT's underlying Vite/Rollup build to **avoid leading underscores** in chunk filenames. The key is to override the Rollup output pattern for chunks (using Vite's `build.rollupOptions.output.chunkFileNames`). By supplying a pattern that prefixes chunk names with a non-underscore character (e.g. adding a `"c-"` prefix), the `_commonjsHelpers...` file will be renamed to something Chrome accepts.

### Steps:

1. **Open your** `wxt.config.ts` and use the `vite` override to adjust Rollup output settings via `defineConfig`. For example:

```
import { defineConfig } from 'wxt';

export default defineConfig({
  vite: () => ({
    build: {
      rollupOptions: {
        output: {
          // Prepend "c-" to chunk names to avoid leading "_" in filenames
          chunkFileNames: 'assets/c-[name]-[hash].js',
        },
      },
    },
  })
})
```

```
  })  
});
```

Here we put chunks in an `assets/` subdirectory and use `c-[name]-[hash].js` as the naming pattern. This ensures any chunk (including Rollup's CommonJS helper) will start with "c-" instead of "\_".

1. **Rebuild your extension** with this updated config. The problematic file should now be named something like `c-_commonjsHelpers.<hash>.js` (the chunk's base name still contains `_commonjsHelpers`, but now it's prefixed with "c-", so the **filename no longer begins with "\_"**).
2. **Load the unpacked extension in Chrome.** It should load without the previous error, since no file or folder starts with an underscore now.

This approach is officially supported via Vite/Rollup config. In fact, a Vite maintainer specifically recommended setting a custom `chunkFileNames` pattern to fix issues with leading-underscore filenames <sup>2</sup>, and users confirmed it resolves the problem <sup>3</sup>. By leveraging WXT's ability to customize the underlying Vite config, you avoid any hacks and ensure Chrome's filename restrictions are satisfied.

## References

- Chrome extension error on underscore-prefixed files <sup>1</sup> (Chrome disallows filenames starting with "\_").
- Vite discussion on avoiding underscore-prefixed chunks <sup>2</sup> <sup>3</sup> (solution by customizing `chunkFileNames` in Rollup output).

---

<sup>1</sup> Extension installed from Chrome Webstore throws error "Package is invalid. Details: 'Cannot load extension with file or directory name \_metadata...'" - Stack Overflow  
<https://stackoverflow.com/questions/23918539/extension-installed-from-chrome-webstore-throws-error-package-is-invalid-detail>

<sup>2</sup> <sup>3</sup> vite build,avoid generating files starting with underscores. · vitejs vite · Discussion #18532 · GitHub  
<https://github.com/vitejs/vite/discussions/18532>