

Antigravity Docs to PDF — VS Code Extension

A lightweight VS Code extension that exports Antigravity AI agent markdown artifacts (`walkthrough.md`, `implementation_plan.md`, `task.md`) to high-fidelity PDFs. It uses the user's existing Chrome/Edge installation via `puppeteer-core` to render Mermaid diagrams, GitHub-style alerts, and code blocks with pixel-perfect accuracy.

Proposed Changes

Project Scaffolding

[NEW] `{}` package.json

- VS Code extension manifest with `activationEvents`, `contributes.commands`, and `contributes.configuration`
- Two commands: `antigravity.exportAll` and `antigravity.exportCurrentFile`
- Two settings: `antigravity.brainPath` and `antigravity.exportPath`
- Dependencies: `puppeteer-core`, `markdown-it`, `chrome-launcher`
- Dev dependencies: `@types/vscode`, `typescript`, `@vscode/vsce`, `esbuild`

[NEW] `{}` tsconfig.json

- TypeScript configuration targeting ES2020, module `Node16`, strict mode
- Output to `dist/`

[NEW] `JS` esbuild.js

- Bundle script to produce a single `dist/extension.js` with external `vscode` module
- Copies static assets (`styles.css`, `template.html`) to `dist/`

[NEW] `VSCode` .vscodeignore

- Ignore `src/`, `node_modules/` (except production deps), `tsconfig.json`

Core Extension

[NEW] `TS` src/extension.ts

- `activate(context)`: registers both commands
- `exportAll`: reads `brainPath` setting, globs for `task.md`, `implementation_plan.md`, `walkthrough.md` across all conversation dirs, invokes `PdfGenerator` for each
- `exportCurrentFile`: takes active editor file, invokes `PdfGenerator`
- Progress notification via `vscode.window.withProgress`
- Home directory `~` expansion helper

Browser Detection

[NEW] `TS` src/browserFinder.ts

- Uses `chrome-launcher` to find Chrome installation path
 - Manual fallback paths for Edge on macOS / Windows / Linux
 - Returns the first valid executable path or throws an actionable error
-

Markdown Rendering Pipeline

[NEW] `TS` src/markdownRenderer.ts

- Creates a `markdown-it` instance with HTML enabled
 - **Custom plugin: GitHub Alerts** — regex-based rule that transforms `> [!NOTE]`, `> [!TIP]`, `> [!IMPORTANT]`, `> [!WARNING]`, `> [!CAUTION]` blockquotes into styled `<div>` elements with SVG icons and accent colors
 - **Custom plugin: Mermaid** — detects ````mermaid` fenced code blocks and wraps content in `<div class="mermaid">` instead of `<pre><code>`
 - **Custom plugin: Carousel** — detects ````carousel` blocks and renders them as styled slide containers (simplified for PDF — shows all slides stacked)
 - Syntax highlighting for code blocks via `markdown-it`'s built-in highlight or a lightweight highlighter
-

PDF Generator

[NEW] `TS` src/PdfGenerator.ts

- `generate(mdFilePath, outputPath)` method:
 1. Read markdown file from disk
 2. Pass through `markdownRenderer` to get HTML body
 3. Read `template.html` and `styles.css`, inject body + styles
 4. Inject `mermaid.min.js` via CDN (<https://cdn.jsdelivr.net/npm/mermaid/dist/mermaid.min.js>)
 5. Launch `puppeteer-core` with browser path from `browserFinder`
 6. Set page content with `page.setContent(html, { waitUntil: 'networkidle0' })`
 7. **Wait for Mermaid:** `page.waitForFunction` that checks all `.mermaid` divs contain rendered `<svg>` elements, with a 15-second timeout
 8. Call `page.pdf()` with A4 format, margins, `printBackground: true`
 9. Close browser
 - Error handling with user-friendly messages via `vscode.window.showErrorMessage`
-

Static Assets

[NEW] `TS` src/styles.css

- GitHub-Light / Antigravity Clean aesthetic
- System font stack (`-apple-system, BlinkMacSystemFont, "Segoe UI", Helvetica, Arial`)
- Typography: proper heading hierarchy, line-height, letter-spacing

- Code blocks: rounded corners, syntax coloring, line numbers
- GitHub alerts: colored left-border, background tint, SVG icon for each type (NOTE=blue, TIP=green, IMPORTANT=purple, WARNING=yellow, CAUTION=red)
- Mermaid diagrams: centered, `page-break-inside: avoid`
- Tables: zebra striping, rounded headers
- Print-specific rules: `@page` margins, `page-break-inside: avoid` for code/diagrams/alerts

[NEW] — src/template.html

- Minimal HTML5 shell: `<head>` with style injection point, `<body>` with content injection point
- Mermaid initialization script with `startOnLoad: true`, `theme: 'default'`

Extension Assets

[NEW] 🖼 images/icon.png

- A simple extension icon (generated)

[NEW] 📄 README.md

- Usage instructions, screenshots placeholder, configuration docs

Verification Plan

Automated Tests

1. **TypeScript compilation:** Run `npx tsc --noEmit` to verify type correctness

```
cd /Users/clarencenguyen/projects/ag-docs-to-pdf && npx tsc --noEmit
```

2. **Esbundle bundle:** Run the build script and verify `dist/extension.js` is produced

```
cd /Users/clarencenguyen/projects/ag-docs-to-pdf && node esbuild.js && ls -la dist/
```

3. **VSIX packaging:** Run `vsce package` to verify the extension can be packaged

```
cd /Users/clarencenguyen/projects/ag-docs-to-pdf && npx @vscode/vsce package --no-dependencies
```

Manual Verification

1. **Install the VSIX in VS Code:** `Extensions > Install from VSIX...`
2. **Open a workspace** and run `Cmd+Shift+P > Antigravity: Export Current File` on a markdown file containing Mermaid diagrams and alerts
3. **Verify the output PDF:**
 - Mermaid diagrams are rendered as crisp SVGs (not blank)

- GitHub alerts have proper coloring and icons
- Code blocks have syntax highlighting
- No content is split awkwardly across pages
- The overall aesthetic matches a premium GitHub-light look