

# 软著鉴别材料整理器 软件说明书

版本：1.0

目录

目录 ..... 1

正文 ..... 2

软件概述 ..... 2

功能列表 ..... 3

    核心功能：程序鉴别材料生成 ..... 3

    扩展功能：AI 辅助文档生成 ..... 4

技术特点 ..... 5

运行环境要求 ..... 5

    最终用户环境 ..... 5

    开发/构建环境 ..... 6

使用说明 ..... 6

    一、 获取与安装 ..... 6

    二、 生成程序鉴别材料（核心流程） ..... 6

    三、 使用 AI 生成文档鉴别材料（可选） ..... 7

    四、 开发者调试与打包 ..... 7

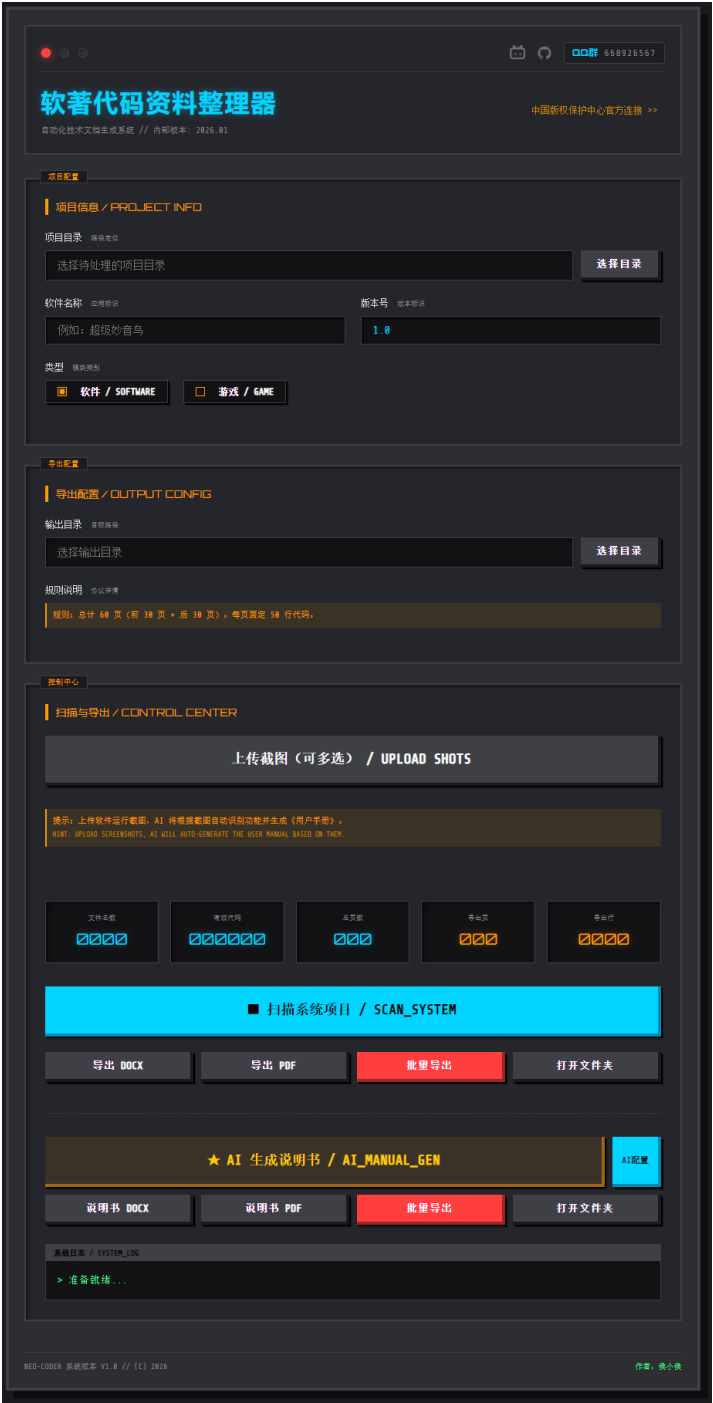
## 正文

### 软件概述

软著鉴别材料整理器 (Softwork Code Organizer) 是一款专为软件著作权登记流程设计的桌面应用程序。其核心目标是自动化、规范化地生成软件著作权登记所需的“程序鉴别材料”（即源代码部分），帮助开发者、企业或代理机构高效、准确地准备申报材料。

本软件遵循中国版权保护中心对程序鉴别材料的常见要求（如前 30 页后 30 页，每页 50 行有效代码），通过智能扫描用户指定的项目源代码目录，自动过滤无效内容，按规范格式生成可直接用于提交的 DOCX 或 PDF 文档。它显著减少了人工整理代码、计算行数、调整格式的重复性工作，降低了材料准备的门槛和出错率。

# 软著鉴别材料整理器软件 V1.0



## 功能列表

### 核心功能：程序鉴别材料生成

#### 1. \*\*智能项目扫描\*\*

- \* \*\*自动识别\*\*：支持扫描 JavaScript/TypeScript、Python、Java、C/C++、C#、Go、Rust、PHP、Swift、Kotlin 等主流编程语言的源代码文件。
- \* \*\*目录排除\*\*：自动跳过与著作权鉴别无关的常见目录，如 `node\_modules`、

`dist`、`build`、`.git`、`vendor` 等，确保抽取的代码具有代表性。

- \* **实时统计**：扫描完成后，在界面清晰展示扫描到的总文件数、总有效代码行数、预估总页数及计划导出的页数/行数。

## 2. **规范的代码处理与抽样**

- \* **有效行定义**：准确定义“有效代码行”为非空且非纯注释的行，确保抽取的每一行都具有实际意义。

- \* **智能抽样规则**：

- \* 若有效代码总页数超过 60 页，则自动抽取前 30 页与后 30 页的代码。

- \* 若总页数不足或等于 60 页，则导出全部代码。

- \* **连续性保证**：代码按文件路径排序后连续分页，不受单个文件边界限制，确保页码连续。对于超长代码行，会自动换行处理并计入行号统计。

- \* **容量不足处理**：当项目总有效代码行不足 3000 行（即 60 页）时，软件会在前 30 页区域循环填充代码，确保第 3000 行（即最后一页的最后一行）恰好是某个源文件的结束行，方便人工核对材料的完整性。

## 3. **专业文档导出**

- \* **DOCX 导出**：生成符合软著申报习惯的 Microsoft Word 文档。文档包含规范的页眉（格式为[软件名称] 软件/游戏软件 V[版本号]）、页脚（“第 X 页”样式的页码），并**利用 Word 内置的连续行号功能**为所有代码行编号，这与版权中心受理时的常见格式高度一致。

- \* **PDF 导出**：通过调用本机已安装的 Microsoft Word 应用程序，将生成的 DOCX 文档高质量地转换为 PDF 格式，完美保留行号、分页和版面样式。

- \* **灵活导出选项**：支持单独导出 DOCX 或 PDF，也支持同时导出两种格式。

## 扩展功能：AI 辅助文档生成

1. **AI 智能分析**：在用户授权并提供 API 密钥后，软件可调用大语言模型（如 GPT-4）分析项目目录结构和关键源代码文件。

2. **自动生成说明书**：基于 AI 对项目的分析，自动撰写符合软著要求的“文档鉴别材料”（即软件操作说明书或设计说明书）初稿。

3. **图文混排支持**：允许用户上传软件界面截图，AI 可将截图与文字描述智能结合，生成图文并茂的说明书草稿。

4. **文档导出**：支持将 AI 生成的说明书内容导出为格式规整的 DOCX 或 PDF 文档。

未找到匹配截图：AI 生成说明书区

## 技术特点

1. **跨平台桌面架构**：基于 **Electron** 框架开发，使用 Web 技术（HTML, CSS, JavaScript）构建用户界面，并打包为原生应用程序，实现了对 Windows、macOS 和 Linux 三大桌面操作系统的完美支持。
2. **清晰的多进程架构**：
  - \* **主进程**（`src/main.js`）：负责窗口管理、系统原生对话框（文件选择、保存）、核心业务逻辑的 IPC 通信调度、自动更新及数据持久化（通过 `electron-store`）。
  - \* **渲染进程**（`src/renderer/`）：负责呈现用户界面、处理用户交互，并通过预加载脚本（`preload.js`）安全地与主进程通信。
  - \* **共享模块**（`src/shared/`）：将扫描器（`scanner.js`）、文档生成器（`exportDocx.js`）、AI 处理器（`aiScanner.js`）等核心业务逻辑独立封装，供主进程和渲染进程通过 IPC 调用，保证代码复用和架构清晰。
3. **专业文档处理**：使用 **docx** 库在 Node.js 环境中以编程方式构建复杂的 Word 文档，精确控制页眉、页脚、段落样式、行号、字体等所有格式细节，无需依赖 Office 软件即可生成标准 DOCX 文件。
4. **用户友好的交互设计**：
  - \* 采用状态指示灯（红/黄/绿）直观反映应用当前状态（空闲、忙碌、成功、错误）。
  - \* 提供详尽的扫描统计信息和导出预览。
  - \* 支持一键打开输出目录，方便用户查看生成的文件。
5. **自动化发布与更新**：集成 **electron-builder** 和 **electron-updater**，支持通过 GitHub Actions 工作流进行自动化跨平台打包，并内置静默检查与提示更新机制，便于用户获取新版本。

## 运行环境要求

### 最终用户环境

- \* **操作系统**：
  - \* Windows 7 及以上 (推荐 Windows 10/11)
  - \* macOS 10.15 (Catalina) 及以上
  - \* Linux (支持主流发行版，如 Ubuntu, Fedora, 以 Appliance 格式运行)

- \* **\*\*硬件\*\***: 具备运行上述操作系统的基本硬件配置。
- \* **\*\*可选依赖（用于 PDF 导出）\*\***: 本地需安装 **\*\*Microsoft Word\*\*** (Office 365 或独立版本均可)。若无 Word, 将无法使用 PDF 导出功能, 但不影响 DOCX 文件的生成。

## 开发/构建环境

- \* **\*\*Node.js\*\***: 版本 16.x 或更高。
- \* **\*\*npm\*\*** 或 **\*\*yarn\*\*** 包管理器。
- \* **\*\*操作系统\*\***: 建议在目标打包平台 (如 Windows 下打包 Windows 应用) 进行本地构建, 或使用配置好的 CI/CD 流水线 (如 GitHub Actions) 进行跨平台构建。

## 使用说明

### 一、获取与安装

1. 访问项目的 GitHub Releases 页面。
2. 根据您的操作系统下载对应的安装包:
  - \* **\*\*Windows\*\***: 下载 `.exe` 安装程序。
  - \* **\*\*macOS\*\***: 根据芯片架构下载 `.dmg` (Apple Silicon) 或 `.dmg` (Intel) 文件。
  - \* **\*\*Linux\*\***: 下载 `.ApplImage` 文件。
3. 运行安装包并按提示完成安装 (Windows/macOS) 或为 ApplImage 文件添加可执行权限后直接运行 (Linux)。

### 二、生成程序鉴别材料 (核心流程)

1. **\*\*配置软件信息\*\***: 启动应用, 在 “软件名称” 和 “软件版本” 输入框中填写您要申报的软件信息。
2. **\*\*选择项目路径\*\***: 点击 “选择项目目录” 按钮, 定位到您需要生成鉴别材料的软件项目根文件夹。
3. **\*\*选择输出目录\*\***: 点击 “选择输出目录” 按钮, 指定生成文档的保存位置。
4. **\*\*扫描项目\*\***: 点击 “扫描项目” 按钮。软件将自动分析项目结构, 过滤文件, 并统计出有效代码信息。扫描结果将显示在界面中央。
5. **\*\*导出文档\*\***:
  - \* 点击 “导出 DOCX” 直接生成 Word 文档。
  - \* 若本机已安装 Microsoft Word, 可点击 “导出 PDF” 生成 PDF 文档。

\* 点击“同时导出”可一次性生成两种格式的文档。

6. **\*\*查看结果\*\***：导出完成后，可点击“打开输出目录”快速查看生成的规范文档。

### 三、使用 AI 生成文档鉴别材料（可选）

1. **\*\*配置 AI\*\***：点击界面上的 AI 功能相关按钮，在弹出的配置窗口中填入您的大模型 API 密钥和基础 URL（例如使用 OpenAI 或国内兼容 API 的模型）。

未找到匹配截图：AI 配置弹窗

2. **\*\*上传截图\*\***：点击“选择截图”按钮，上传您的软件界面截图，用于辅助 AI 生成图文并茂的说明书。

未找到匹配截图：上传图片区

3. **\*\*生成说明书\*\***：确保已选择项目路径，点击“AI 生成说明书”按钮。软件将分析项目并调用 AI 生成说明书草稿，内容会显示在界面的文本区域。

4. **\*\*编辑与导出\*\***：您可以在文本区域内对 AI 生成的内容进行修改和润色。完成后，使用“导出说明书 DOCX/PDF”按钮将最终文档保存到本地。

### 四、开发者调试与打包

1. **\*\*环境准备\*\***：克隆项目代码，在根目录下执行 ``npm install`` 安装所有依赖。

2. **\*\*启动开发\*\***：执行 ``npm start`` 命令，启动 Electron 开发窗口。

3. **\*\*打包应用\*\***：执行 ``npm run dist`` 命令，将在 ``dist/`` 目录下生成当前系统平台对应的安装包。跨平台构建建议使用项目预置的 GitHub Actions 工作流。