

目录

- [快速上手](#)
 - [第一次打开先做什么](#)
 - [1. 确认游戏路径](#)
 - [2. 扫描模组](#)
 - [3. 整理启用列表](#)
 - [4. 自动排序](#)
 - [5. 保存加载顺序](#)
 - [6. 启动游戏](#)
 - [推荐的日常流程](#)
 - [新装一批 Mod 后](#)
 - [更新或删除 Mod 后](#)
 - [导入别人分享的列表后](#)
 - [常见问题](#)
 - [扫描后没有看到工坊 Mod](#)
 - [排序后还存在警告](#)
 - [保存后游戏里没变化](#)
 - [不确定下一步点什么](#)
 - [注意事项](#)
- [环境管理与路径说明](#)
 - [先按你的用法选择](#)
 - [Steam 正版普通使用](#)
 - [多个独立存档或配置环境](#)
 - [不通过 Steam 启动，但想使用已有工坊文件](#)
 - [使用 SteamCMD 下载到管理器库](#)
 - [一张图理解路径关系](#)
 - [环境里保存什么](#)
 - [全局路径和环境路径](#)
 - [跟随当前环境](#)
 - [所有环境共用](#)
 - [关键路径怎么填](#)
 - [游戏本体目录](#)

- [用户数据目录](#)
 - [本地模组目录](#)
 - [创意工坊目录](#)
 - [管理器下载模组路径](#)
 - [Steam 程序路径](#)
 - [SteamCMD 目录](#)
 - [工具模组目录](#)
- [自动识别怎么工作](#)
- [Mod 是怎么进入游戏的](#)
 - [本地 Mod](#)
 - [Steam 工坊 Mod](#)
 - [管理器模组](#)
 - [重复 Mod 怎么选](#)
- [链接部署模式](#)
 - [增量部署](#)
 - [完全重建](#)
- [启动方式怎么选](#)
 - [直接启动游戏本体](#)
 - [Steam 客户端启动](#)
 - [Steam URL 启动](#)
 - [不同来源本体的启动选择](#)
- [创建环境快捷方式](#)
 - [本体直接启动快捷方式](#)
 - [Steam 官方 AppID 快捷方式](#)
 - [Steam 本地库快捷方式](#)
- [常见问题](#)
 - [自动识别不到游戏](#)
 - [创意工坊路径校验失败](#)
 - [SteamCMD 路径提示不能包含中文](#)
 - [本地 Mods 里多了_Link_...](#)
 - [启动前链接同步失败](#)
 - [Steam 启动和工坊链接挂载怎么选](#)
- [重要注意事项](#)
- [工坊与下载来源说明](#)
 - [先按场景选择](#)
 - [希望 Steam 自动更新管理模组](#)

- [希望手动管理模組的更新](#)
 - [工坊合集导入](#)
 - [模組只在 Git 仓库发布](#)
- [库存枢纽页面](#)
- [Steam 订阅](#)
- [SteamCMD 下载](#)
- [创意工坊检索](#)
- [合集订阅管理](#)
- [Git 仓库订阅](#)
 - [Source 模式](#)
 - [Release 模式](#)
 - [zip 直链](#)
- [网络访问和下载加速 \(仅限中国大陆网络环境\)](#)
 - [按问题选择设置](#)
 - [GitHub 镜像怎样选](#)
 - [软件更新怎样处理多个下载来源](#)
 - [代理开关影响哪些功能](#)
 - [动态 Hosts 怎样使用](#)
 - [检测、保存与排查](#)
- [更新检查](#)
- [推荐用法](#)
 - [普通 Steam 玩家](#)
 - [整合包爱好者](#)
 - [测试新 Mod](#)
- [常见问题](#)
 - [订阅了但列表里没有](#)
 - [SteamCMD 下载成功但游戏没加载](#)
 - [Git 仓库订阅后不能用](#)
 - [工坊项目显示失效](#)
- [注意事项](#)
- [模組列表与加载顺序说明](#)
 - [列表里的 Mod 从哪里来](#)
 - [四类列表怎么理解](#)
 - [停用列表](#)
 - [启用列表](#)
 - [临时列表](#)

- [禁用列表](#)
- [常用操作](#)
 - [启用和停用](#)
 - [双击](#)
 - [拖拽排序](#)
 - [多选](#)
 - [扩选 \(划动框选\)](#)
 - [多选次序问题](#)
 - [右键菜单](#)
 - [联锁功能](#)
- [搜索和筛选](#)
 - [基础写法](#)
 - [与 / 或 逻辑](#)
 - [常用字段](#)
 - [判断类字段](#)
 - [排除搜索](#)
 - [搜索和排序的注意事项](#)
 - [禁用列表里的搜索](#)
- [分组、标签、颜色和备注](#)
 - [分组](#)
 - [标签](#)
 - [颜色](#)
 - [备注](#)
- [问题提示怎么看](#)
 - [从哪里看问题](#)
 - [红色错误](#)
 - [黄色警告](#)
 - [蓝色提示](#)
 - [推荐排查顺序](#)
 - [什么时候可以忽略](#)
- [重复、共存和缺失](#)
 - [重复包名](#)
 - [共存版本](#)
 - [缺失项](#)
- [保存和备份](#)
- [常见问题](#)

- [我停用了 Mod, 文件会被删吗?](#)
 - [拖拽后为什么游戏里没变?](#)
 - [为什么同一个 Mod 出现多个版本?](#)
 - [为什么自动排序后还有问题?](#)
- [自动排序与规则说明](#)
 - [自动排序做了什么](#)
 - [排序原理简化理解](#)
 - [自动排序不能保证什么](#)
 - [规则来源](#)
 - [原生规则](#)
 - [用户规则](#)
 - [社区规则](#)
 - [创意工坊规则](#)
 - [动态规则](#)
 - [生效优先级](#)
 - [权重怎么理解](#)
 - [贴靠行为](#)
 - [语言包贴紧前置](#)
 - [普通 Mod 贴紧依赖](#)
 - [贴靠不是强制相邻](#)
 - [冲突规则怎么处理](#)
 - [问题提示怎么处理](#)
 - [缺少依赖](#)
 - [排序错误](#)
 - [不兼容](#)
 - [语言包提示](#)
 - [推荐 workflow](#)
 - [普通排序](#)
 - [创建自己的规则](#)
 - [使用动态规则](#)
 - [常见问题](#)
 - [自动排序是不是越多规则越好?](#)
 - [社区规则和用户规则冲突怎么办?](#)
 - [修改规则后为什么列表没变化?](#)
 - [忽略警告安全吗?](#)
 - [注意事项](#)

- [备份与导入导出说明](#)
 - [加载顺序备份](#)
 - [导入加载顺序](#)
 - [导出加载顺序](#)
 - [加载备份前要确认什么](#)
 - [跨环境查看备份](#)
 - [分享码](#)
 - [推荐清单导出](#)
 - [软件数据导入导出](#)
 - [模组实体包导入导出](#)
 - [推荐流程](#)
 - [大改排序前](#)
 - [导入别人列表](#)
 - [换电脑](#)
 - [常见问题](#)
 - [备份会保存 Mod 文件吗?](#)
 - [导入后为什么有缺失项?](#)
 - [ModsConfig.xml 和推荐清单有什么区别?](#)
 - [软件数据包和模组整合包怎么选?](#)
 - [注意事项](#)
- [贴图优化说明](#)
 - [先说结论](#)
 - [贴图优化到底做了什么](#)
 - [各种方案有什么区别](#)
 - [游戏原生压缩画质要不要开](#)
 - [建议怎么选](#)
 - [开启的好处](#)
 - [可能的代价](#)
 - [生成范围怎么选](#)
 - [DDS](#)
 - [ZSTD](#)
 - [缩放比例](#)
 - [最小清晰度](#)
 - [自动回退](#)
 - [不缩放、遮罩贴图和超范围](#)
 - [Mipmap](#)

- [清理已生成贴图](#)
- [加载时间为什么不一定明显变短](#)
- [显存预估为什么和实际不完全一样](#)
- [推荐用法](#)
 - [只想稳妥优化](#)
 - [大型模组列表，想平衡画质和性能](#)
 - [显存很紧张](#)
 - [磁盘空间紧张](#)
- [常见问题](#)
 - [生成 DDS 后可以删除 PNG 吗？](#)
 - [画面变糊怎么办？](#)
 - [ZSTD 生成了但游戏没用怎么办？](#)
 - [为什么有些图片没有缩放？](#)
 - [我应该什么时候重新生成？](#)
- [注意事项](#)
- [日志排错与 AI 诊断说明](#)
 - [什么时候看日志](#)
 - [日志中心里有什么](#)
 - [基本排错流程](#)
 - [1. 先复现问题](#)
 - [2. 找到关键错误](#)
 - [3. 结合 Mod 名和包名](#)
 - [4. 调整后再测试](#)
 - [一键分析](#)
 - [手动选择日志分析](#)
 - [AI 助手怎么问更有效](#)
 - [工具权限](#)
 - [辅助工具模组](#)
 - [文件内容搜索](#)
 - [常见排错场景](#)
 - [启动阶段报错](#)
 - [进入存档后报错](#)
 - [大量 PatchOperation 报错](#)
 - [某个贴图或资源加载失败](#)
 - [隐私和安全](#)
 - [常见问题](#)

- [AI 说的可疑 Mod 一定对吗?](#)
- [日志太长发不出去怎么办?](#)
- [系统日志和游戏日志该看哪个?](#)
- [改完后怎么确认修好了?](#)
- [注意事项](#)

快速上手

这份文档适合第一次使用 RimCrow 的玩家。目标很简单：让软件识别你的游戏和 Mod，整理加载顺序，保存后启动游戏。

如果你只想先跑通流程，按下面四步走：

```
扫描 -> 自动排序 -> 保存 -> 启动游戏
```

第一次打开先做什么

1. 确认游戏路径

先确认 RimCrow 找到的是 RimWorld 安装根目录，不是 exe 文件本身，也不是 `Data` 或 `RimWorldWin64_Data`。

Steam 版通常在：

```
Steam库/steamapps/common/RimWorld
```

用户数据目录应选择 RimWorld 的用户数据根目录，里面通常有：

```
Config/  
Saves/
```

路径、环境、Steam 启动和工坊链接的详细说明见 [环境管理与路径说明](#)。

2. 扫描模组

路径确认后，先点击“扫描”。扫描会读取当前环境下能看到的本地 Mod、工坊 Mod、管理器模组和相关信息。

普通扫描适合日常使用。发现列表明显不对、刚手动改过文件、或者怀疑缓存没有刷新时，再尝试更彻底的扫描。

3. 整理启用列表

主界面通常分成两侧：

- 停用列表：你拥有但当前不加载的 Mod。
- 已启用列表：游戏实际会读取的加载顺序。

常用操作：

- 双击 Mod：在启用和停用之间切换。
- 拖拽 Mod：调整位置或移动到另一侧。
- 批量选择：`Ctrl+左键` 多选、`Shift+左键` 连选、`Ctrl+A` 全选、从序号长按鼠标划动扩选。
- 右键菜单：执行更多操作，例如打开目录、编辑信息、处理来源。

红色或黄色提示一般表示依赖、顺序、兼容性或缺失问题。启动前建议先处理红色错误。

模组列表的详细说明见 [模组列表与加载顺序说明](#)

4. 自动排序

点击“自动排序”后，RimCrow 会根据 Mod 原生依赖、社区规则、用户规则、动态规则等信息重新整理启用列表。

自动排序不是“保证完美”的按钮。它更适合先给出一个合理基础顺序；排序后仍要看问题提示和差异，再决定是否手动微调。

自动排序和规则的详细说明见 [自动排序与规则说明](#)。

5. 保存加载顺序

拖拽、启停和自动排序后的变化，只有保存后才会真正写入当前环境的加载顺序文件。

保存前建议：

- 看一下是否还有红色错误。
- 确认当前环境是不是你想改的环境。

- 重要列表可以先保留备份。

6. 启动游戏

保存后再启动游戏。RimCrow 会按当前环境的设置启动 RimWorld，并在需要时先同步管理器链接。

如果启动前链接同步失败，不会继续启动。先按提示检查路径、权限和 `_Link_...` 链接状态。

推荐的日常流程

新装一批 Mod 后

1. 扫描。
2. 把需要的 Mod 移到已启用列表。
3. 自动排序。
4. 处理缺失依赖、顺序错误和兼容性提示。
5. 保存。
6. 启动游戏测试。

更新或删除 Mod 后

1. 扫描。
2. 查看缺失、已删除、重复或共存提示。
3. 必要时重新自动排序。
4. 保存。
5. 启动游戏。

导入别人分享的列表后

1. 在备份页导入文件或分享码。
2. 先看差异和缺失项。
3. 使用补订阅、补下载或手动处理缺失项。
4. 确认列表合理后再加载到当前环境。
5. 保存并启动。

备份和导入导出说明见 [备份与导入导出说明](#)。

常见问题

扫描后没有看到工坊 Mod

先检查创意工坊目录是否填写正确，路径一般是：

```
steamapps/workshop/content/294100
```

如果你开启了 Steam 启动，工坊内容主要由 Steam 处理；如果你不通过 Steam 启动但想读取已有工坊目录，需要开启“使用创意工坊 Mod”。

排序后还存在警告

这很常见。自动排序只能根据已有规则处理问题，没有规则或信息不足时仍需要用户确认。

红色错误优先处理。黄色警告可以结合 Mod 说明、规则来源和实际使用情况判断。

保存后游戏里没变化

先确认：

- 保存的是当前环境。
- 游戏没有同时开着旧实例。
- RimWorld 读取的用户数据目录和当前环境一致。
- 没有手动给启动参数添加错误的 `-savedatafolder`。

不确定下一步点什么

按主流程走：扫描、自动排序、保存、启动。遇到问题时，先看列表顶部的问题汇总，再看具体 Mod 的提示。

注意事项

- 第一次整理大型列表时，不建议一次处理太多不熟悉的高级功能。
- 重要列表建议保存前保留备份。
- 不要直接删除数据库或用户数据目录来“修复问题”。
- 分享日志、路径或导出包前，注意里面可能包含本地路径和 Mod 清单。

环境管理与路径说明

这份文档说明 RimCrow 里的“环境”、常见路径、工坊模组接入、管理器模组接入、启动方式和快捷方式。你不需要理解底层实现，只要知道每个路径该填什么、每个开关适合什么时候打开。

先按你的用法选择

如果你不确定怎么设置，先看这一节。

Steam 正版普通使用

适合想让 Steam 负责工坊订阅、更新和加载的玩家。

- 游戏本体目录：选择 Steam 安装的 RimWorld。
- Steam 程序路径：保持自动识别，识别不到再手动选择 Steam。
- 优先使用 Steam 启动：开启。
- 使用创意工坊 Mod：关闭。
- 使用管理器模组：只在你需要加载管理器下载库里的 Mod 时开启。

这类用法最接近 RimWorld 和 Steam 的默认行为。

多个独立存档或配置环境

适合测试不同 Mod 列表、不同版本配置或整合包。

- 多个环境可以共用同一个游戏本体目录。
- 每个环境使用独立用户数据目录。
- 需要继承当前配置时，新建环境时复制当前 `Config` 和 `Saves`。
- 不要手动添加 `-savedatafolder`，RimCrow 会自动处理。

环境隔离主要隔离的是存档、设置、加载顺序和日志等用户数据，不会自动复制游戏本体。

不通过 Steam 启动，但想使用已有工坊文件

适合本机已经有 Steam 工坊目录，但当前环境不想交给 Steam 启动。

- 创意工坊目录：填写 `steamapps/workshop/content/294100`。
- 优先使用 Steam 启动：关闭。
- 使用创意工坊 Mod：开启。
- 用 RimCrow 启动游戏，让它在启动前同步链接。

这种方式只是读取本机已有工坊文件，不会让非 Steam 游戏连接 Steam 工坊服务。

使用 SteamCMD 下载到管理器库

适合想把工坊 Mod 下载到 RimCrow 管理器模组目录，不依赖 Steam 客户端订阅列表的情况。

- SteamCMD 目录：放到无中文路径，例如 `D:/Tools/steamcmd`。
- 管理器下载模组路径：建议放到 NTFS 磁盘。
- 不要把管理器下载模组路径设成 Steam 官方工坊目录。
- 下载后开启“使用管理器模组”，再扫描和启动。

SteamCMD 下载和 Steam 客户端订阅不是一回事，更新和卸载口径也不同。

一张图理解路径关系

RimWorld 启动后主要读取两类位置：

```
游戏本体目录/  
  RimWorldWin64.exe  
  Mods/  
  
用户数据目录/  
  Config/  
    ModsConfig.xml  
  Saves/
```

RimCrow 做的事可以简单理解为：

- 选择当前环境使用哪个游戏本体目录。

- 选择当前环境使用哪个用户数据目录。
- 启动前把需要加载的管理器模组或工坊模组链接到 `Mods`。
- 启动游戏时自动带上当前环境需要的参数。

所以排查路径问题时，先分清楚：这是游戏程序位置、用户数据位置，还是 Mod 文件来源位置。

环境里保存什么

RimCrow 里的“环境”是一套 RimWorld 运行配置。它主要包含：

- 游戏本体目录：RimWorld 程序所在目录。
- 用户数据目录：存档、设置、加载顺序、日志等数据所在目录。
- 启动参数：随游戏一起传给 RimWorld 的额外参数。
- 运行开关：是否优先使用 Steam 启动、是否使用创意工坊 Mod、是否使用管理器模组。

默认环境通常对应系统原本的 RimWorld 用户数据目录。新建环境如果没有手动指定用户数据目录，RimCrow 会放到管理器数据目录下：

```
data/profiles/<环境ID>
```

新目录会在保存或激活环境时自动创建 `Config` 和 `Saves`。

全局路径和环境路径

有些设置是“当前环境”的，有些是“所有环境共用”的。

跟随当前环境

- 游戏本体目录
- 用户数据目录
- 本地模组目录
- 启动参数
- 优先使用 Steam 启动
- 使用创意工坊 Mod

- 使用管理器模组

所有环境共用

- Steam 程序路径
- SteamCMD 目录
- 创意工坊目录
- 管理器下载模组路径
- 工具模组目录

这意味着：切换环境会改变当前游戏本体、用户数据和运行开关，但不会为每个环境单独保存一套 SteamCMD 或工坊目录。

关键路径怎么填

游戏本体目录

这里应选择 RimWorld 的安装根目录，不要选择 exe 文件本身，也不要选择 `Data` 或 `RimWorldWin64_Data`。

常见可执行文件：

- Windows: `RimWorldWin64.exe` 或 `RimWorldWin.exe`
- Linux: `RimWorldLinux` 或 `RimWorldLinux.x86_64`
- macOS: `RimWorldMac.app`

Steam 安装的 RimWorld 通常在：

```
Steam库/steamapps/common/RimWorld
```

Steam 版、复制出来的 Steam 旧版、GOG / Epic / 官网版本都可以作为游戏本体目录。不同来源的本体主要区别在于启动方式和工坊加载方式，后面的“启动方式怎么选”会继续说明。

用户数据目录

这里应选择 RimWorld 用户数据根目录，不要选择 `Config`、`Saves` 或 `Config/ModsConfig.xml`。

正确结构通常是：

```
用户数据目录/  
  Config/  
    ModsConfig.xml  
  Saves/
```

常见默认位置：

- Windows: `%USERPROFILE%/AppData/LocalLow/Ludeon Studios/RimWorld by Ludeon Studios`
- macOS: `~/Library/Application Support/RimWorld`
- Linux: `~/.config/unity3d/Ludeon Studios/RimWorld by Ludeon Studios`

如果目录还不存在，只要父目录存在且可写，RimCrow 可以在保存或激活环境时自动创建。

本地模组目录

本地模组目录由游戏本体目录决定，一般是：

```
RimWorld/Mods
```

这个目录不需要单独填写。多个环境如果共用同一个游戏本体，也会共用同一个本地 `Mods` 目录。

创意工坊目录

这里填写 Steam 下载 RimWorld 工坊 Mod 的目录。RimWorld 的 Steam AppID 是 `294100`，常见结构是：

```
Steam库/steamapps/workshop/content/294100
```

这个目录由 Steam 管理。订阅、更新和卸载仍由 Steam 负责，**RimCrow 不会自动更新或删除内部文件**，你可以手动更新或删除。另外不建议把它设置为管理器下载库使用。

如果 `294100` 目录还没生成，先在 Steam 里订阅或下载一个 RimWorld 工坊项目。

管理器下载模组路径

这里存放 RimCrow 通过 SteamCMD 或 Git 等方式下载和管理的 Mod。默认在管理器目录的 `mods` 下，也可以改到其它位置。

不要把它设置成 Steam 官方工坊目录。两者混在一起后，Steam 和 RimCrow 都可能以为自己在管理同一批文件，后续排查会很麻烦。

如果你把管理器下载模组路径设成游戏本体的 `Mods` 目录，建议关闭“使用管理器模组”，避免重复链接。

Steam 程序路径

Steam 程序路径用于 Steam 启动、订阅、工坊状态读取等功能。一般可以自动识别；识别不到时再手动选择 Steam 安装目录，即 Steam 程序本体所在目录。

如果路径无效，Steam 启动和 Steam 工坊相关能力可能不可用，但直接启动游戏本体仍可使用。

SteamCMD 目录

SteamCMD 是 Valve 提供的命令行下载工具。路径应选择包含 `steamcmd.exe` 或 `steamcmd.sh` 的文件夹。

注意：

- 路径不要包含中文。
- Windows 下建议放在 NTFS 磁盘。
- SteamCMD 会按自己的标准目录下载：

```
SteamCMD目录/steamapps/workshop/content/294100
```

RimCrow 会尽量把这个下载目录和“管理器下载模组路径”关联起来，让 SteamCMD 按官方结构工作，同时让用户在管理器库里看到下载结果。

工具模组目录

工具模组目录存放 RimCrow 自带或辅助用途的 Mod，路径固定，普通用户一般不用改。只有启用对应工具模组开关后，它们才会参与扫描和运行时链接部署。

自动识别怎么工作

RimCrow 不会默认全盘乱扫，而是优先读取比较稳定的位置：

- Steam 安装信息。
- Steam 库配置 `libraryfolders.vdf`。
- RimWorld 的 `appmanifest_294100.acf`。
- 各平台常见默认安装目录。
- 各平台默认用户数据目录。

这样速度快，也不容易误扫到无关目录。但如果你移动过 Steam 库、手动复制过游戏、删除过 Steam 配置文件，自动识别可能不完整，可以手动纠正。

识别失败时，直接手动选择路径即可。路径校验通过后，后续功能会按你手动填写的路径工作。

Mod 是怎么进入游戏的

RimWorld 天然只认识游戏本体目录下的 `Mods`，以及 Steam 启动时由 Steam 处理的工坊内容。RimCrow 根据你的开关决定是否在启动前做链接部署。

本地 Mod

放在 `RimWorld/Mods` 里的 Mod 会被游戏直接读取。RimCrow 不需要额外链接。

Steam 工坊 Mod

有两种方式：

方式	适合情况	关键开关
Steam 启动	Steam 正版玩家，想让 Steam 负责工坊	开启“优先使用 Steam 启动”
链接挂载	不通过 Steam 启动，但想读取已有工坊文件	关闭“优先使用 Steam 启动”，开启“使用创意工坊 Mod”

这两个方式不要同时使用。开启 Steam 启动时，RimCrow 会关闭工坊链接挂载，避免 Steam 和管理器重复处理同一批工坊 Mod。

管理器模组

管理器下载模组路径里的 Mod 不会被 RimWorld 自动读取。开启“使用管理器模组”后，RimCrow 会在启动前把需要运行的管理器 Mod 链接到当前游戏本体的 Mods 目录。

最终游戏看到的是：

- Mods 目录里的真实本地 Mod。
- RimCrow 创建的 _Link_... 链接。

重复 Mod 怎么选

如果多个来源的模组有相同 PackageId，RimCrow 会选择优先级更高的一份，避免同一个 Mod 被多份来源同时挂入。大致优先级是：

DLC / 官方内容 > 本地 Mods > 管理器模组 > Steam 工坊模组 > 工具模组

链接部署模式

链接部署只管理 _Link_ 前缀的链接，不会主动删除普通本地 Mod 文件夹。

增量部署

默认推荐。它会保留已经正确的链接，只删除失效、错误或当前环境不再需要的链接，再补齐缺失链接。

完全重建

会先删除所有 `_Link_` 前缀的旧链接，再按当前环境重新创建。

普通用户建议保持增量部署。只有在手动改过 `_Link_` 目录、切换环境后怀疑链接残留、或链接状态明显错乱时，再尝试完全重建。

启动方式怎么选

直接启动游戏本体

RimCrow 直接运行 RimWorld 可执行文件，并附带当前环境的启动参数。常见参数是：

- `-popupwindow` 无边框窗口模式
- `-quicktest` 快速测试

适合：

- 非 Steam 版。
- 不想通过 Steam 启动。
- 需要用链接方式加载管理器模组或工坊目录。
- 测试不同用户数据环境。

启动前如果需要同步链接，RimCrow 会先处理链接。同步失败时不会继续启动，避免带着错误 Mod 状态进入游戏。

Steam 客户端启动

RimCrow 通过 Steam 启动 RimWorld，常见形式是：

```
steam.exe -applaunch 294100 ...参数
```

这种方式可以携带环境启动参数，适合需要 `-savedatafolder` 的环境。

Steam URL 启动

Steam URL 形式类似：

```
steam://run/294100
```

它调用方式更加简单通用，即使不设置 Steam 路径也能使用，但无法携带额外启动参数。只有当前环境没有启动参数时，才建议使用 Steam URL。

不同来源本体的启动选择

Steam 当前主安装目录里的 RimWorld，最适合直接交给 Steam 启动。这样 Steam 会按订阅状态处理官方工坊 Mod，RimCrow 只负责当前环境的路径和启动参数。

有些环境使用的不是 Steam 当前主安装目录，例如复制出来的旧版本 Steam 本体、备份版，或整合包自带的游戏目录。只要本机 Steam 账号拥有 RimWorld，并且这个本体能被 Steam 正常接管，就可以尝试通过 Steam 启动。这样可以一定程度上让这类环境按 Steam 方式运行，直接使用 Steam 官方工坊内容；是否稳定可用，取决于本体文件、Steam 客户端和账号状态。

GOG、Epic、官网等其它平台版本也可以作为独立环境直接启动。它们通常不能直接使用 Steam 官方工坊服务；如果你本机已经有 Steam 工坊目录，可以关闭“优先使用 Steam 启动”，开启“使用创意工坊 Mod”，让 RimCrow 用链接方式把已有工坊文件挂到本地 Mods 目录。

创建环境快捷方式

环境快捷方式会复用当前环境的启动方式和启动参数，不需要你手写命令。不同环境会按当前启动方式生成不同类型的快捷方式。

本体直接启动快捷方式

这种快捷方式直接指向 RimWorld 可执行文件，并带上当前环境的启动参数。

适合：

- 非 Steam 版或不想通过 Steam 启动的环境。
- 使用链接方式加载管理器模组或已有工坊文件的环境。
- 需要明确启动某个游戏目录的测试环境。

它不会让 Steam 接管启动，也不会让 Steam 自动处理官方工坊订阅。

Steam 官方 AppID 快捷方式

这种快捷方式指向 Steam 程序，并使用 `-applaunch 294100` 启动不同环境的 RimWorld。

适合 Steam 当前主安装目录里的 RimWorld 为基础创建的环境。它可以携带当前环境的启动参数，也可以让 Steam 按官方方式处理工坊内容。

如果 Steam 路径无效，RimCrow 会回退创建“游戏本体”快捷方式，并给出提示。

Steam 本地库快捷方式

这种方式适合不在 Steam 当前主安装目录里的 Steam 本体，例如复制出来的旧版本 Steam 本体、备份版或整合包目录。

在 Windows 上，RimCrow 可以把这个本体作为 Steam 的“非 Steam 游戏”加入本地库，再创建对应的桌面快捷方式。流程大致是：

1. 先完全退出 Steam，避免配置被覆盖。
2. RimCrow 写入或更新 Steam 的 `shortcuts.vdf`。
3. 重新启动 Steam。
4. 等 Steam 生成稳定的 `steam://rungameid/...`。
5. 创建桌面 URL 快捷方式。

它和 Steam 官方 AppID 快捷方式的区别是：官方 AppID 启动的是 Steam 当前管理的 RimWorld；本地库方式启动的是你指定路径下的那个 RimWorld 本体。

可能生成的快捷方式类型：

- Windows 普通启动： `.lnk`
- Windows Steam URL： `.url`
- Linux： `.desktop`
- macOS： `.command`

根据环境不同，快捷方式可能指向：

- 游戏本体可执行文件。
- Steam 程序，并带 `-applaunch 294100` 参数。
- `steam://rungameid/...` 这类 Steam URL。

如果多个环境共用同一个游戏本体目录，快捷方式只能保证启动目标和参数正确；`Mods` 目录里的链接状态仍以最近一次由 RimCrow 启动或同步的环境为准。

常见问题

自动识别不到游戏

先确认 Steam 或系统里确实能找到 RimWorld。仍识别不到时，手动选择 RimWorld 安装根目录。不要选到 exe 文件、`Data` 或 `RimWorldWin64_Data`。

创意工坊路径校验失败

检查路径是否是 `steamapps/workshop/content/294100`。如果 `294100` 目录不存在，先在 Steam 里订阅或下载一个 RimWorld 工坊项目。

SteamCMD 路径提示不能包含中文

把 SteamCMD 放到纯英文路径，例如 `D:/Tools/steamcmd`。中文路径容易导致 SteamCMD 下载或更新失败。

本地 Mods 里多了 `_Link_...`

这是 RimCrow 创建的运行时链接，用来让游戏读取管理器模组、工具模组或链接挂载的工坊模组。它不是完整复制出来的 Mod。正常情况下不要手动改名或移动。

启动前链接同步失败

优先检查：

- 游戏本体 `Mods` 目录是否存在且可写。
- 管理器模组目录或创意工坊目录是否还存在。
- 当前磁盘是否支持目录连接。
- 杀毒软件或文件管理器是否占用了 `_Link_` 目录。
- 是否手动删除或改名为 RimCrow 创建的链接。

必要时切换为“完全重建”链接部署模式，再重新扫描和启动。

Steam 启动和工坊链接挂载怎么选

能通过 Steam 正常启动并使用订阅工坊时，优先用 Steam 启动。只有在不通过 Steam 启动、但又需要读取本机工坊目录里的文件时，再开启工坊链接挂载。

重要注意事项

- 环境隔离主要隔离用户数据目录，不会自动复制游戏本体。
- 共用同一个游戏本体的多个环境，会共用同一个本地 `Mods` 目录。
- 不要手动添加 `-savedatafolder`，多环境管理会自动处理。
- 管理器链接部署只管理 `_Link_` 前缀链接，不会删除普通本地 Mod 文件夹。
- Steam 官方工坊目录由 Steam 管理，不建议当成管理器下载库。
- SteamCMD 下载目录和管理器下载模组路径可能通过目录连接关联，看到 Junction 或链接是正常现象。
- 路径越接近 Steam 和 RimWorld 的标准结构，自动识别越稳定；手动搬家、便携 Steam、跨平台兼容层路径更可能需要手动填写。

工坊与下载来源说明

这份文档说明 RimCrow 里几种常见 Mod 获取方式：Steam 订阅、SteamCMD 下载、合集导入、Git 仓库订阅和 zip 直链。

先记住一个区别：

- Steam 订阅：由 Steam 客户端管理，管理与加载 Steam Workshop 创意工坊的模组。
- SteamCMD 下载：通过 SteamCMD 将创意工坊模组下载到 RimCrow 的管理器模组库，绕过 Steam 客户端限制。
- Git / 直链：适合不在创意工坊的以仓库发布的 Mod。

先按场景选择

希望 Steam 自动更新管理模组

优先用 Steam 订阅。

适合：

- Steam 正版玩家。
- 希望 Steam 自动更新工坊 Mod。
- 不想手动管理下载目录。

设置建议：

- 优先使用 Steam 启动：开启。
- 使用创意工坊 Mod：关闭。
- 工坊订阅、更新和卸载交给 Steam。

希望手动管理模组的更新

用管理器的 SteamCMD 下载模组到 RimCrow 管理器下载模组路径。

适合：

- 想手动管理工坊 Mod 的更新变动。
- 不想依赖当前 Steam 订阅列表。
- 想和 Git / 直链 来源一起集中管理。

下载后需要开启“使用管理器模组”，游戏才会通过链接方式读取这些 Mod。

工坊合集导入

在合集订阅管理中导入目标合集链接或工坊id。

适合：

- 别人给了 Steam 合集链接或合集 ID。
- 你想快速查看合集包含哪些 Mod。
- 希望随时查看合集内容是否更新。
- 你想补齐缺失项，再决定是否套用顺序。

模组只在 Git 仓库发布

用 Git 仓库订阅。

适合：

- Mod 没有上传创意工坊。
- 作者在 GitHub、GitLab 或 GitGud 发布源码。
- 作者通过 Release 发布打包版本。

库存枢纽页面

点击“库存枢纽”会进入工作区，顶部有几个主要页面：

- 全库数据管理：查看工坊、本地和管理器库库存。
- 创意工坊检索：搜索工坊项目，查看详情、依赖、截图和作者信息。
- 合集订阅管理：导入 Steam 合集，查看合集内项目和缺失状态。
- Git 仓库订阅：添加 Git / 直链 来源，查看订阅状态和部署记录。

Steam 订阅

Steam 订阅会把工坊项目加入你的 Steam 订阅列表。下载、更新和卸载主要由 Steam 客户端负责。

适合：

- 正常使用 Steam 版 RimWorld。
- 希望跟随工坊自动更新。
- 希望保持官方行为。

注意：

- 需要 Steam 已登录并能访问工坊。
- 订阅后 Steam 不一定立刻下载完成。
- 取消订阅不等于 RimCrow 一定马上看不到文件，仍要看 Steam 是否已清理本地文件。

SteamCMD 下载

SteamCMD 是独立下载工具。RimCrow 可以用它把工坊项目下载到管理器模组库。

适合：

- 想独立保存 Mod 文件。
- 想让管理器库集中管理下载内容。
- 想绕开 Steam 客户端订阅列表做下载。

注意：

- SteamCMD 目录不要包含中文。
- 下载目录建议在 NTFS 磁盘。
- SteamCMD 下载通常不等于 Steam 客户端订阅。
- 下载后需要扫描，列表才会更新。
- 使用这些 Mod 前，当前环境要开启“使用管理器模组”。

SteamCMD 路径和管理器模组路径的关系见 [环境管理与路径说明](#)。

创意工坊检索

创意工坊检索可以按 Mod 名、包名或工坊 ID 搜索。

结果区通常会显示：

- Mod 名称和工坊 ID。
- 包名、作者、版本信息。
- 订阅状态。
- 简介、依赖、截图和相关项目。

常用动作：

- 订阅到 Steam。
- 取消订阅。
- 下载到管理器。
- 打开网页工坊页面。
- 查看依赖、同作者项目或相关项目。

增强模式可以获取更完整的工坊信息；关闭时会更多依赖本地缓存和公开接口，结果可能不完整。

合集订阅管理

合集页适合处理别人分享的 Steam 工坊合集。

推荐流程：

1. 粘贴合集 URL 或输入合集 ID。
2. 选择导入的合集记录。
3. 查看包含哪些 Mod。
4. 检查哪些已订阅、已安装或缺失。
5. 批量订阅或下载缺失项。
6. 最后再决定是否把合集顺序应用到当前列表。

不要一导入就直接覆盖当前启用列表。先看差异和缺失项更稳。

Git 仓库订阅

Git 仓库订阅适合没有上传工坊，或作者主要在代码托管平台发布的 Mod。

支持常见公开仓库：

- GitHub
- GitLab
- GitGud

也支持部分 zip 直链来源。

Source 模式

Source 模式会获取源码分支内容。

适合：

- 作者没有发布 Release。
- 你想跟进测试版或最新提交。
- 仓库结构本身就是可用 Mod。

风险：

- 源码分支可能不稳定。
- 有些仓库需要构建后才是可用 Mod。

Release 模式

Release 模式会获取作者发布的打包版本。

适合：

- 作者有正式 Release。
- 你想使用更稳定的版本。
- 不想处理源码目录结构。

如果仓库没有 Release，这个模式不可用。

zip 直链

zip 直链适合推荐清单或第三方来源里的打包文件。

注意：

- zip 来源依赖链接稳定性。
- 链接失效后需要换来源。
- 下载后仍要扫描和确认 Mod 结构是否正确。

网络访问和下载加速（仅限中国大陆网络环境）

入口：**设置** → **网络连接**。网络正常时不必调整；遇到访问慢或失败时，按问题选择对应选项，不需要全部开启。

按问题选择设置

遇到的问题	可以尝试	需要注意
GitHub 项目页面、文件或下载包访问慢、打不开	GitHub 镜像源	只帮助访问节点支持的 GitHub 内容，不改善 Steam 等其他网站
多个外部网站无法访问，且你已有可用的代理服务	启用代理服务	按代理工具提供的信息填写协议、主机地址和端口；没有可用服务时，单独打开开关无效
提示找不到服务器或域名解析失败	动态 Hosts	只帮助查找网站地址，不能修复服务器故障或代理连接失败

GitHub 镜像怎样选

在“GitHub 镜像源”的“镜像节点”中选择：

- **GitHub 直连**：不用镜像，访问 GitHub 原地址；如果配置了代理，仍会遵循代理设置。
- **具体节点**：优先使用所选节点；节点不支持对应内容或访问失败时，会尝试 GitHub 原地址。
- **自动选择镜像**：先打开“启用动态镜像节点”，等待获取和检测，再选择此项，让软件从当前可用节点中选择。

需要自定义节点时，使用选择器旁的新增、编辑或删除按钮。只使用可信节点；带账号凭据的请求和私有资源不会交给公共镜像。

软件自身更新采用独立策略：先访问 GitHub 原地址，失败后自动尝试内置的 `gh-proxy.org`，不跟随这里选择的镜像节点；即使选择“GitHub 直连”，软件更新仍保留该备用路径。它只读取公开发布信息和安装包，不向公共镜像发送账号凭据，且仍遵循应用代理设置。

软件更新怎样处理多个下载来源

在“关于项目”中点击“检查更新”。软件会汇总多个发布平台的结果，只展示最高适用版本的一份更新说明；展开“来源详情”可查看各站的检查结果。部分来源失败时仍可使用已发现的更新，未发现更新也不代表无法访问的来源一定没有新版本。

点击“下载更新”后可在任务栏查看进度，失败时自动尝试同版本备用地址。下载完成后可以稍后安装；继续安装时需依次确认安装和重启，任意一次取消都不会执行安装。稍后可从“关于项目”的“查看更新”重新打开。

代理开关影响哪些功能

- **启用代理服务**是应用代理总开关，用于普通查询、文件下载和更新检查等请求；镜像地址与原地址都会遵循此设置。“不走代理的域名”用于排除特定网站。
- **为 AI 请求使用代理**单独控制 AI 是否使用这里配置的代理，需要同时打开总开关；关闭 AI 开关不影响普通请求。
- **为 SteamCMD 使用代理**需要同时打开总开关，但不保证能解决 SteamCMD 的所有连接或下载问题。
- 关闭总开关只是不再使用这里配置的代理，**不等于强制直连**；外部代理环境仍可能生效。这里的设置不修改其他程序的代理，也不保证控制 Steam 客户端或内置浏览器的连接。
- 软件更新使用蓝奏云备用来源时，会直接连接，不使用这里配置的代理或环境代理。

动态 Hosts 怎样使用

1. 打开“动态 Hosts”，检查“Hosts 来源”中已启用的地址；通常无需修改，确需添加时只使用可信来源。
2. 等待获取结果。显示“Hosts 已获取，保存设置后生效”时，点击底部的“应用并保存配置”；获取失败时可检查网络和来源后点击“重新获取”，已有映射会保留。
3. 需要定期更新时，开启“自动更新动态 Hosts”并设置更新间隔。它会在软件启动时检查是否已到更新时间。

Hosts 不限于 GitHub 或 Steam，能帮助访问哪些网站取决于来源包含哪些域名。明确知道正确 IP 时，也可填写“自定义 Hosts 映射”；同一域名以手动填写的映射为准。

默认只影响软件自身的后台请求，不自动应用到内置浏览器。“**将 Hosts 映射应用到系统**”会**修改系统文件，并影响其他程序，可能需要管理员权限，通常无需开启**。确需使用时，保存后按提示确认是否成功，再刷新或重开浏览器；通过代理访问时，这些映射不一定生效。

检测、保存与排查

调整后可点击右上角“检测访问”，查看当前表单配置下各站点的结果；将鼠标停在结果上可查看连接详情。**检测不会保存或应用表单中的配置，动态 Hosts 的获取预览也不会更改当前生效的映射**。要验证刚获取的动态映射，请先保存，再检测。

- 检测延时不是大文件下载速度；一个站点检测成功，也不代表它的所有页面和文件都能访问，更不代表 AI 或 SteamCMD 已通过检测。
- 确认可用后点击“应用并保存配置”；不想保留本次设置时点击“放弃修改”。
- 工坊浏览器打不开时，先确认外部网络能访问 Steam 工坊，再检查浏览器使用的网络环境；不要仅凭后台检测成功判断浏览器也能访问。页面能打开但图片缺失时，还需确认图片来源也能访问。
- 仍然失败时，记录失败的功能和提示，按 [日志排错与 AI 诊断说明](#) 查看日志，再有针对性地调整，不必反复叠加镜像、代理和 Hosts。

更新检查

RimCrow 可以显示工坊、SteamCMD 和 Git 来源的状态变化，但不同来源的更新时间口径不同。

常见状态：

- 已安装：本地能找到文件。
- 已订阅：Steam 订阅列表里存在。
- 有变动：发生了明显文件变动或更新的模组。
- 有更新：远端或记录显示可能有新版本。
- 已删除：可能在游戏中或其它第三方因素导致的模组缺失或删除。
- 缺失：记录存在，但本地没有有效文件，通常仍处于订阅状态，可能为工坊下架或 Steam 同步等原因导致的文件意外删除。

遇到状态不一致时，先重新扫描，再检查来源本身是否可访问，然后进行相应的处理操作，如：重新订阅、重新下载等。

推荐用法

普通 Steam 玩家

- 工坊 Mod 用 Steam 订阅。
- 用 Steam 启动游戏。
- 不把 Steam 工坊目录当成管理器库。

整合包爱好者

- 用合集页导入别人分享的合集。
- 用备份导入导出管理加载顺序。
- 用 SteamCMD 或 Git 订阅补充非 Steam 来源。
- 保留每次大改前的备份。

测试新 Mod

- 先下载到管理器库或单独环境。
- 扫描后只启用少量相关 Mod。
- 自动排序并保存。
- 进入游戏测试，确认稳定后再并入常用环境。

常见问题

订阅了但列表里没有

Steam 可能还没下载完成。先等 Steam 下载结束，再重新扫描。

SteamCMD 下载成功但游戏没加载

检查当前环境是否开启“使用管理器模组”，并确认启动前链接同步成功。

Git 仓库订阅后不能用

优先检查模组文件是否成功下载，可能因网络原因导致下载失败。另外可能是仓库源码不是直接可用的 Mod，或需要 Release 打包版本。优先查看仓库 README 和文件结构。

工坊项目显示失效

可能是项目下架、隐藏、地区访问异常，或本地缓存过旧。先刷新和重新扫描，再尝试打开工坊页面确认。

注意事项

- 不要把 Steam 官方工坊目录和管理器下载模组路径混用。
- 订阅、下载、部署是不同动作，完成一个不代表另一个已经完成。
- 更新前后建议扫描，避免列表使用旧状态。
- 删除或取消订阅前先确认来源，避免删错副本。

模组列表与加载顺序说明

这份文档说明主界面的 Mod 列表怎么用，以及哪些操作会影响游戏实际加载顺序。

简单理解：

- 停用列表：当前不加载，但仍然保留在普通 Mod 库里的 Mod。
- 启用列表：游戏实际加载读取的 Mod 顺序。
- 临时列表：只在管理器里临时整理用，与游戏无关，可视为停用列表。
- 禁用列表：被明确禁止进入游戏的 Mod，游戏里一般看不到。
- 保存：把当前启用顺序写入对应环境的游戏配置。

列表里的 Mod 从哪里来

RimCrow 扫描后会把不同来源的 Mod 放到同一套管理视图里。常见来源包括：

- 本地 Mod：游戏本体 `Mods` 目录里的 Mod。
- Steam 工坊 Mod：Steam 工坊目录里的 Mod。
- 管理器模组：RimCrow 通过 SteamCMD、Git 或直链等方式下载的 Mod。
- 官方内容：游戏本体和 DLC。

不同来源会影响部署方式、更新方式和删除方式。不要只看 Mod 名称，遇到重复或共存时要先看来源。

四类列表怎么理解

停用列表

这里放的是当前环境没有启用的 Mod。它们仍然存在于你的磁盘、工坊库或管理器库里，只是当前加载顺序没有使用。

适合：

- 暂时不用的 Mod。

- 等待测试的 Mod。
- 有冲突但暂时不想删除的 Mod。

注意：

- 停用不等于删除文件。
- 停用的 Mod 仍可能在管理器里被搜索、分组、打标签或再次启用。
- 如果只是今天不用，放停用列表通常比禁用更合适。

启用列表

这里就是当前环境的实际加载顺序。越靠上越早加载，越靠下越晚加载。

常见情况：

- 前置库、框架、官方内容通常靠前。
- 普通内容 Mod 居中。
- 补丁、语言包、性能优化、末端修复通常靠后。

具体顺序仍以 Mod 说明、规则提示和实际测试为准。

临时列表

临时列表在右侧辅助栏里。它像一个“暂存区”，用于临时放一批你正在整理、测试或准备批量移动的 Mod。

适合：

- 从停用列表里先挑出一批候选 Mod。
- 导入列表后临时观察，不马上写进启用列表。
- 做一轮测试前，把待处理 Mod 单独放一边。
- 整理大型列表时，先把一类 Mod 暂时集中起来。

注意：

- 临时列表里的 Mod 不会进入游戏加载顺序，效果更接近停用。
- 临时列表主要是管理器里的整理状态，不要把它当成备份。
- 设置里有“保存临时列表”：开启后会按当前环境恢复；关闭后，保存时会把临时列表里的 Mod 放回停用列表顶部。
- 如果要长期保存一批 Mod 组合，优先使用分组或备份，不要只依赖临时列表。

禁用列表

禁用列表在右侧辅助栏的“已禁用”页。这里放的是你明确不希望游戏读取的 Mod。

适合：

- 已确认会造成冲突，暂时不想让游戏看到的 Mod。
- 多个副本里不想参与当前环境的一份。
- 从冲突处理或库存管理中标记为禁用的 Mod。
- 需要严格控制整合包可见范围的场景。

注意：

- 禁用比停用更强。停用只是“不加载”，禁用是“尽量不让游戏看到它”。
- 已禁用 Mod 可以在禁用列表中筛选、排序、右键解除禁用。
- 禁用列表里的删除、取消订阅会影响真实来源，操作前先确认路径和来源。
- 开启严格禁用模式后，扫描时会按管理器记录保持禁用状态；如果外部把它恢复了，RimCrow 会尝试重新禁用。
- 如果只是测试暂时不用，先放停用或临时列表；只有明确不想让它进入环境时再禁用。

常用操作

启用和停用

- 双击 Mod 可以快速启用或停用。
- 也可以拖拽到另一侧列表。
- 多选后可批量启用或停用。

启用只是把 Mod 加入当前环境的加载列表，不等于复制文件。停用也不会自动删除 Mod 文件。

双击

双击是最快的启停方式：

- 在停用列表双击：加入启用列表。
- 在启用列表双击：移回停用列表。

双击后，管理器会在当前排序规则允许范围内尽量找最合理的位置插入 Mod。另外注意：启用列表变动后及时保存，游戏里才会读取新的启用顺序。

拖拽排序

在已启用列表中拖拽可以调整加载顺序。拖拽后的变化需要保存才会写入配置。

如果你手动调整了顺序，建议再看一次问题提示，避免把依赖项拖到错误位置。

在筛选或视图排序状态下，列表只是在“视觉检阅”。这时通常不能拖拽插入或调整真实顺序，但是可以多选拖出到其它列表。想改实际顺序，可以先将目标 Mod 拖入临时列表，然后清掉筛选和视图排序，再拖回原列表或其它位置。

启用、停用和临时列表的“名称”排列优先使用别名，没有别名时使用原名称；选择“原名”则只按原名称排列，不受别名影响。两种排列都支持升序和降序，只改变显示顺序，不修改游戏的加载顺序。

多选

常用选择方式：

- 单击：选中一个 Mod。
- `Ctrl` + 单击：追加选择或取消选择一个 Mod。
- `Shift` + 单击：从上一次选中的位置到这次点击的位置连续选择。
- `Ctrl` + `Shift` + 单击：在保留已有选择的基础上追加一段连续范围。
- `Ctrl+A`：全选当前可见列表或筛选结果。

批量操作前，先确认当前筛选条件。否则你可能只操作了筛选结果，而不是完整列表。

扩选（划动框选）

列表左侧的序号列或选择触发区支持划动扩选。按住左键从一个 Mod 划到另一个 Mod，会把中间这一段一起选中。

常见用法：

- 选中一段连续 Mod 后批量启用、停用、移动或打标签。
- 在大列表中比一个个 `Ctrl` 点击更快。
- 配合筛选先缩小范围，再划动扩选。

注意：

- 划动扩选按当前可见顺序计算。
- 按住 `Ctrl` 再划动，会在原有选择基础上追加这一段。
- 如果你开始拖动 Mod 卡片本身，可能会进入拖拽排序；想扩选时尽量从序号列或选择触发区开始。

多选次序问题

创建联锁、批量移动这类操作，会参考当前选中项的顺序。不同选择方式的“顺序来源”不一样。

扩选和连选最直观：

- 扩选：也就是划动框选，按当前列表显示顺序选中一段。
- 连选：也就是 `Shift` + 单击，同样按当前列表显示顺序选中一段。

所以用扩选或连选创建联锁时，联锁顺序通常就是你屏幕上看到的上下顺序。即使当前开着筛选或问题过滤，也会按过滤后的可见顺序计算。

`Ctrl` 多选不一样。它是一个个点出来的，点击位置可能前后跳动，所以顺序主要按你的点击次序确定。比如先点第 20 个，再点第 5 个，再点第 12 个，创建联锁时就可能看起来“不按列表上下顺序”。这不是随机，而是因为这次多选本身没有连续范围。

如果你要创建联锁，推荐这样做：

- 一段连续的 Mod：优先用扩选或 `Shift` 连选。
- 分散的 Mod：可以用 `Ctrl` 多选，但创建前最好确认选中顺序。
- 发现联锁顺序不符合预期：解除联锁后，按列表显示顺序重新扩选或连选，再创建联锁。

右键菜单

右键菜单适合处理单个或多个 Mod 的常见动作，例如：

- 打开 Mod 路径。
- 打开工坊页面。
- 编辑别名、标签、备注。
- 处理启用、停用、删除、订阅等动作。
- 进入规则编辑或来源处理。

带删除含义的操作要先确认来源。删除本地文件、删除工坊文件、取消订阅不是同一个动作。

联锁功能

联锁用于把几项必须相邻的 Mod 绑成一组。创建联锁后，选中其中一个成员时，RimCrow 会自动把同一联锁组里的成员一起带上；拖拽或移动时，也会尽量保持这一组的相对顺序。

适合：

- 主 Mod 和专用补丁必须紧贴。
- 一组框架、扩展、兼容补丁需要成套移动。
- 你不想每次手动排序都重新找它们。

常见操作：

- 多选几个 Mod 后，右键创建联锁。
- 选中已有联锁成员后，右键解除联锁。
- 出现“联锁模组缺失”或“联锁排序错误”时，按提示补齐、下载、订阅、剔除失效项或重新排序。

注意：

- 联锁不是依赖规则。它只表达“这几项最好成组相邻”，不能替代前置依赖或兼容性判断。
- 不要把太多无关 Mod 联锁在一起，否则后续排序和移动会变笨重。
- 自动排序仍会参考依赖、规则和权重；联锁只是其中一个约束。

搜索和筛选

主列表有两个输入框，作用不同：

- 上方“搜索定位”：不改变列表内容，只在当前列表中跳到匹配项。左键定位下一个，右键定位上一个。
- 下方“筛选”：缩小当前显示范围，只显示符合条件的 Mod。

筛选后注意看列表数量。很多“找不到 Mod”的情况，其实是筛选条件还开着。

基础写法

常见写法：

- 直接输入关键词：按默认字段模糊搜索。
- `类别:关键词`：只在指定字段里搜索。
- `-类别:关键词`：排除匹配这个字段的 Mod。
- `类别:+` / `类别:-` / `类别:_`：用于判断字段是否有值、无值、未知。
- 多个条件可以选择“与”或“或”逻辑。

默认直接搜索会匹配名称、别名和作者。包名、工坊 ID、标签、来源等需要用字段搜索更准确。

输入时可以用 `Tab` 应用建议。空输入时，建议列表会显示可用字段。

与 / 或 逻辑

多个搜索条件可以选择关系：

逻辑	含义	例子
与	每个条件都满足才显示	<code>tags:性能 + source:workshop</code>
或	满足任意条件就显示	<code>tags:性能 + tags:UI</code>

排查问题时通常用“与”，找同类 Mod 时常用“或”。

常用字段

这些字段适合在主列表里使用：

字段	含义	示例
<code>name</code>	Mod 名称	<code>name:Harmony</code>
<code>alias_name</code>	自定义别名	<code>alias_name:核心</code>
<code>author</code>	作者	<code>author:Oskar</code>
<code>package_id</code>	包名	<code>package_id:ludeon</code>
<code>workshop_id</code>	工坊 ID	<code>workshop_id:2009463077</code>
<code>tags</code>	标签	<code>tags:性能</code>
<code>groups</code>	分组	<code>groups:中世纪</code>
<code>sign_color</code>	颜色标记	<code>sign_color:red</code>
<code>mod_type</code>	类型	<code>mod_type:语言包</code>
<code>source</code>	来源	<code>source:workshop</code>
<code>store</code>	存放位置	<code>store:self</code>
<code>supported_versions</code>	支持版本	<code>supported_versions:1.6</code>
<code>supported_languages</code>	支持语言	<code>supported_languages:zh</code>
<code>multiplayer_compat</code>	联机兼容性	<code>multiplayer_compat:compatible</code>
<code>ignored_issues</code>	已忽略的问题	<code>ignored_issues:缺少依赖</code>

字段名输入不完整时，可以看输入建议。部分字段也会自动生成较短别名，实际以输入框建议为准。

判断类字段

判断类字段适合查 “有没有某种状态”：

字段	含义	示例
<code>last_active</code>	最近是否启用过	<code>last_active:+</code>
<code>coexist_variant</code>	是否存在共存版本	<code>coexist_variant:+</code>
<code>shadow_paths</code>	是否存在禁用副本	<code>shadow_paths:+</code>
<code>replacement</code>	是否存在替代项	<code>replacement:+</code>
<code>save_breaking</code>	是否标记为坏档风险	<code>save_breaking:+</code>

三种值的含义：

- `+`：有值 / 是。
- `-`：无值 / 否。
- `_`：未知 / 空。

例如：

- `replacement:+`：只看存在替代项的 Mod。
- `save_breaking:+`：只看有坏档风险提示的 Mod。
- `coexist_variant:-`：只看没有共存版本的 Mod。

排除搜索

在字段前加 `-` 表示排除。

例子：

- `-tags:测试`：隐藏带“测试”标签的 Mod。
- `-source:workshop`：隐藏工坊来源。
- `-author:unknown`：排除作者里包含 unknown 的项。

排除搜索也可以和其它条件一起用。例如：`tags:性能 + -source:local`，表示只看带“性能”标签、但不是本地来源的 Mod。

搜索和排序的注意事项

- 筛选和排序主要用于查看，不代表真实加载顺序已经改变。
- 筛选或排序开启时，通常不能拖拽改变实际顺序。

- 清掉筛选、恢复默认排序后，看到的才是当前列表的真实顺序。
- 搜索定位不会改变列表内容；筛选会改变可见范围。
- `Ctrl+A`、`Shift` 连选和划动扩选都按当前可见列表计算。

禁用列表里的搜索

禁用列表使用更简单的搜索框，可按名称、包名、工坊 ID 和路径筛选；旁边还能按来源和时间、体积、名称排序。

禁用列表的排序只用于查看，不表示这些 Mod 会进入加载顺序。

分组、标签、颜色和备注

这些功能主要服务于整理，不一定直接改变游戏加载行为。

分组

分组像收藏夹或主题包，适合保存一批相关 Mod，例如“中世纪包”“测试包”“常用 UI”。

把分组拖到启用或停用列表，可以做批量启停。分组本身不是游戏加载顺序文件。

标签

标签适合给 Mod 打长期标记，例如“性能”“种族”“建筑”“待测试”。后续可以用于搜索、筛选和动态规则。

颜色

颜色适合快速识别重点 Mod，比如核心框架、问题 Mod、自己常改的 Mod。

备注

备注适合记录你自己的判断，例如“这个版本别更新”“需要排在某某之后”“多人联机暂不启用”。

问题提示怎么看

列表里的问题提示来自依赖、排序、兼容性、文件缺失、语言包、联锁和联机兼容性检查。它们不是装饰信息，启动前建议至少看一眼启用列表右上角的问题数量。

简单判断：

- 红色错误：优先处理。它很可能影响游戏加载、导致启动失败、进存档报错，或引出更多后续错误。
- 黄色警告：建议确认。一般不会立刻造成严重问题，但可能影响排序、语言显示、联锁完整性或特定玩法。
- 蓝色提示：通常是信息说明，例如依赖已由替代 Mod 满足，主要帮助你理解当前状态。

如果时间有限，先处理红色错误；黄色警告可以根据你是否在意对应功能再决定。

从哪里看问题

常见入口：

- 列表右上角的问题数字：显示当前列表里有多少问题。
- Mod 行上的提示图标：鼠标悬停可查看具体原因。
- 问题筛选：点击问题数量可只看有问题的 Mod。
- 右键菜单：可对部分问题执行忽略、恢复警告、补订阅、补下载或联锁修复。

遇到问题很多时，不要从列表第一行硬看。先点问题汇总筛选，再按红色错误优先处理。

红色错误

红色错误通常表示当前列表已经明显不满足游戏或规则要求。继续启动有较高概率出问题。

常见原因：

- 本地文件缺失或 Mod 解析失败。
- 缺少必要前置。
- 依赖项没有启用。
- 依赖顺序错误，例如必须在前置之后加载却排错了。
- 存在明确不兼容关系。
- Multiplayer 明确不兼容。

建议处理顺序：

1. 先补齐缺失文件和必要前置。
2. 再启用未启用的依赖。
3. 然后自动排序或手动调整顺序。
4. 最后处理明确不兼容项，通常需要二选一、安装兼容补丁，或确认是否有替代方案。

不要用“忽略”来压掉红色错误。只有你非常确定当前环境不受影响时，才考虑忽略。

黄色警告

黄色警告通常表示“可能不理想”或“信息不足”。它一般不会像红色错误那样直接导致游戏启动失败，但仍值得确认。

常见原因：

- 可能缺少推荐前置。
- 语言包指向不明确。
- 当前语言缺少语言包，或语言包对应主 Mod 未启用。
- 排序建议不完全满足。
- Mod 支持版本和当前游戏版本不匹配。
- 连锁组缺成员，或连锁成员没有紧挨着。
- Multiplayer 兼容性较差或未知。
- 来源信息不足，需要人工判断。

建议处理方式：

- 和依赖、排序有关：先自动排序，再看是否还存在。
- 和语言包有关：如果你不需要这个语言，可以暂时不处理；如果要中文或其它语言显示，建议补齐或启用对应语言包。
- 和版本有关：检查 Mod 说明页、更新状态和游戏版本。
- 和连锁有关：把连锁组成员放回相邻位置，或删除已经失效的连锁项。
- 和联机兼容有关：只有玩 Multiplayer 时才需要重点处理；单机玩家可以按需确认。

黄色警告可以忽略，但建议只忽略你已经理解原因的问题。忽略后它不会继续打扰你，不代表问题被修好了。

蓝色提示

蓝色提示通常用于解释当前列表状态，不一定需要操作。

常见情况：

- 某个依赖已由替代 Mod 满足。
- 某些兼容性信息只是提醒你注意。

如果游戏能正常运行，蓝色提示通常可以保留。

推荐排查顺序

日常整理列表时，可以按这个顺序来：

1. 扫描，保证列表状态最新。
2. 点击问题汇总，只看有问题的 Mod。
3. 先处理红色错误。
4. 自动排序，再看剩余问题。
5. 处理你在意的黄色警告。
6. 保存前再看一次问题数量。
7. 启动游戏测试。

每次不要同时改太多。尤其是大型列表，最好一次处理一类问题，确认变化后再继续。

什么时候可以忽略

适合忽略的情况：

- 你已经看过 Mod 说明，确认这个警告对当前玩法无影响。
- 规则库提示较保守，但你实际测试正常。
- 语言包、联机兼容等提示与你当前需求无关。

不建议忽略的情况：

- 缺少必要前置。
- 依赖没有启用。
- 明确不兼容。
- 文件缺失。

- 启动或进存档已经出现红字、卡加载、坏档等问题。

如果不确定，就先不要忽略。保留提示能帮助后续排查。

重复、共存和缺失

重复包名

多个来源出现相同 PackageId 时，RimCrow 会识别出重复或共存状态。不要只按名称判断，因为不同文件夹可能是同一个 Mod 的不同副本。

共存版本

有时你会同时拥有本地版和工坊版。RimCrow 会尽量显示来源差异，并在运行时选择更合适的一份。

需要长期保留多个版本时，建议做好备注，避免以后忘记为什么保留。

缺失项

缺失项通常来自导入列表、旧备份或别人分享的加载顺序。它表示列表里有这个包名，但当前环境没有找到对应 Mod 文件。

可以尝试补订阅、补下载，或手动移除缺失项。

保存和备份

列表变化不会自动写入游戏配置。你需要点击保存。

保存前建议：

- 处理红色错误。
- 确认当前环境正确。
- 对重要列表保留备份。
- 如果刚导入外部列表，先看差异再加载。

更多说明见 [备份与导入导出说明](#)。

常见问题

我停用了 Mod，文件会被删吗？

不会。停用只是从当前环境加载顺序里移除，不等于删除文件或取消订阅。

拖拽后为什么游戏里没变？

需要保存。拖拽和自动排序只是界面里的变更，保存后才写入加载顺序文件。

为什么同一个 Mod 出现多个版本？

通常是本地、工坊、管理器库里有多个副本。先看来源，再决定保留、禁用、删除或转本地。

为什么自动排序后还有问题？

自动排序依赖已有规则和可识别的 Mod 信息。没有规则、规则冲突或 Mod 信息不足时，仍需要你手动判断。

自动排序与规则说明

这份文档说明 RimCrow 的自动排序、规则来源和问题提示。它适合已经完成扫描，并准备整理加载顺序的用户。

先记住一句话：自动排序能帮你得到更合理的基础顺序，但不能保证所有 Mod 组合都一定正确。

自动排序做了什么

自动排序会读取当前启用列表、Mod 依赖信息和各类规则，然后重新排列 Mod 顺序。

它主要处理：

- 前置依赖应排在被依赖 Mod 前面。
- 明确要求“加载在某个 Mod 之后 / 之前”的规则。
- 置顶、置底和权重倾向。
- 语言包、兼容补丁、性能优化等常见顺序建议。

排序完成后，仍建议查看红色错误和黄色警告。

排序原理简化管理解

可以把自动排序理解成三步：

1. 先看必须满足的关系：例如前置依赖、必须在谁前面、必须在谁后面。
2. 再看大致位置倾向：例如置顶、置底、权重靠前或靠后。
3. 最后做局部整理：在不破坏前面规则的前提下，让相关 Mod 尽量靠近。

也就是说，自动排序不是简单按名称或订阅时间排列。它会先保证重要关系，再考虑“看起来更顺”的位置。

举个简单例子：

A 是框架，B 依赖 A，C 是 B 的语言包

排序时通常会让 A 在 B 前面。如果开启语言包贴靠，C 会尽量跟在 B 附近，而不是被单独甩到很远的位置。

如果某些规则互相打架，RimCrow 会尽量保留更重要、更可信的规则，并把无法同时满足的规则提示出来。

自动排序不能保证什么

自动排序不能替你判断所有 Mod 的真实兼容性。

这些情况仍需要人工确认：

- Mod 作者没有写完整依赖信息。
- 社区规则还没有覆盖这个 Mod。
- 两个 Mod 的兼容关系只在说明页里提到。
- 同一 Packageld 有多个来源副本。
- 你使用的是整合包、测试版或旧版 Mod。

规则来源

RimCrow 会综合多种规则来源。不同来源可以共同参与排序和问题检测。

原生规则

来自 Mod 自己的 About.xml、依赖声明和基础信息。

这类规则通常最可靠，因为它来自 Mod 文件本身。但有些 Mod 写得不完整，仍可能需要其它规则补充。

用户规则

由你自己为某个 Mod 写的规则。适合长期固定自己的使用习惯。

常见用途：

- 指定某个 Mod 必须在另一个 Mod 前后。
- 给某个 Mod 设置权重。
- 忽略你确认无影响的提示。
- 修正社区规则没有覆盖的组合。

社区规则

来自社区维护的排序规则库。它可以补充大量常见 Mod 的排序建议。

注意：

- 社区规则是建议，不代表对所有环境都绝对正确。
- 规则库需要更新，旧缓存可能缺少新 Mod。
- 如果规则和你的实际测试冲突，可以用用户规则覆盖或调整。

创意工坊规则

来自创意工坊离线数据库和相关依赖信息。它可以帮助识别工坊项目之间的关系。

如果开启“作为强依赖”，工坊规则会更严格地参与依赖判断。普通用户不确定时，先保持默认。

动态规则

动态规则更像自动化条件：

如果 Mod 满足某些条件 -> 执行某种排序动作

适合批量处理有共同特征的 Mod，例如按标签、类型、作者或名称规则整体前移、后移、置顶或置底。

生效优先级

规则中心里可以调整规则来源优先级。优先级会影响规则冲突时更相信哪一类来源。

简单理解：

- 越靠上，优先级越高。
- 用户规则通常适合放在较高位置，用来表达你自己的最终选择。
- 不确定时不要频繁调整全局优先级，先处理具体 Mod 的问题更稳。

权重怎么理解

权重用于表达排序倾向。数值越小，越靠前；数值越大，越靠后。

常见理解：

- 置顶：强制靠前。
- 置底：强制靠后。
- 权重设定：直接给一个目标区间。
- 权重偏移：在原本位置倾向上前移或后移。

权重适合表达“这一类 Mod 大概应该靠前或靠后”，不适合替代明确依赖关系。

简单说：

- 依赖和前后规则更像“必须这样排”。
- 权重更像“没有硬性冲突时，尽量往这里排”。
- 置顶、置底比普通权重更强，适合少数确实需要固定在两端的 Mod。

不要给太多 Mod 随手设置置顶或置底。两端规则太多时，排序结果反而更难理解。

贴靠行为

贴靠是自动排序后的局部整理。它的目的不是改变规则，而是让有关联的 Mod 看起来更集中。

相关开关在设置中控制。关闭时，排序仍会按依赖、规则和权重处理，只是不会额外做贴靠整理。

语言包贴紧前置

开启后，语言包会尽量跟在它对应的主 Mod 或最后一个已启用前置后面。

这样做的效果是：

- 语言包更容易和主 Mod 一起检查。
- 不容易出现语言包全部堆在列表底部，离主 Mod 很远的情况。
- 语言包较多时，列表会更容易看懂。

注意：

- 如果找不到对应主 Mod，语言包会保持原本的默认靠后位置。
- 如果贴过去会破坏依赖或其它排序规则，RimCrow 不会强行移动。
- 如果语言包归属识别不准，可以手动调整顺序，或写用户规则指定关系。

普通 Mod 贴紧依赖

开启后，普通 Mod 会尽量靠近它最后一个已启用的直接依赖。

这样做的效果是：

- 同一系列或有关联的 Mod 更容易聚在一起。
- 列表不会只按大范围权重分散排列。
- 检查“这个 Mod 依赖谁”时更直观。

注意：

- 这个贴靠主要在相同权重范围内移动，不会随便跨过重要分区。
- 置顶、置底、语言包等特殊位置不会被普通贴靠强行打乱。
- 如果前后规则不允许靠近，RimCrow 会保留更安全的顺序。

贴靠不是强制相邻

贴靠的意思是“尽量靠近”，不是“必须挨着”。

如果中间还有必须插入的依赖、补丁、框架或规则要求，相关 Mod 之间仍可能隔着其它 Mod。这通常不是排序失败，而是为了避免破坏更重要的加载关系。

冲突规则怎么处理

有时不同规则会形成闭环，导致任何顺序都不可能完全满足。

例如：

```
A 必须在 B 前面
B 必须在 C 前面
C 又必须在 A 前面
```

这种情况下，不管怎么排，至少都会违反其中一条规则。

RimCrow 会先给出一个当前能满足最多规则的顺序，然后提示“排序规则冲突”。在冲突视图中：

- 红色规则线表示本次被忽略的规则。
- 其它规则线表示当前排序仍然保留的规则。
- 右侧会显示规则来源，方便判断该调哪一条。

处理建议：

- 如果是用户规则写错了，直接修改或删除用户规则。
- 如果是动态规则范围太宽，收窄条件或暂时关闭这条动态规则。
- 如果是社区规则、工坊规则与你的环境不符，可以关闭对应 Mod 的该来源规则。
- 如果是原生规则，它来自 Mod 文件本身，不能单独禁用，只能调整其它规则或手动处理。

看到冲突提示不代表自动排序完全失败。它表示 RimCrow 已经排出了一个可用的当前结果，但有至少一条规则无法同时满足，建议你确认一下。

问题提示怎么处理

缺少依赖

表示当前启用列表缺少某个必要或建议前置。

处理方式：

- 如果你确实需要这个 Mod，先补齐依赖。
- 如果这是可选依赖，确认无影响后可忽略。
- 如果缺失项来自导入列表，尝试补订阅或补下载。

排序错误

表示当前顺序不满足依赖或规则要求。

处理方式：

- 先尝试自动排序。
- 如果仍不对，手动拖拽调整。
- 长期需要固定时，写用户规则。

不兼容

表示规则认为两个 Mod 不能同时使用，或需要你确认兼容补丁。

处理方式：

- 先查看 Mod 说明或工坊页面。
- 只保留其中一个。
- 安装兼容补丁。
- 你确认无影响时，再选择忽略或写规则。

语言包提示

语言包通常需要跟随它对应的主 Mod。RimCrow 会尽量识别语言包归属，但有些语言包信息不完整。

处理方式：

- 确认主 Mod 是否已启用。
- 让语言包排在合适位置。
- 如果识别错了，可以在规则里手动指定。

推荐 workflow

普通排序

1. 扫描。

2. 启用需要的 Mod。
3. 点击自动排序。
4. 查看红色错误。
5. 手动处理无法自动解决的问题。
6. 保存。

创建自己的规则

适合反复遇到同一个排序问题时使用。

1. 找到相关 Mod。
2. 打开规则编辑。
3. 添加加载前后、权重或忽略规则。
4. 保存规则。
5. 重新自动排序并检查结果。

不要为了临时测试写太多规则。一次性的顺序调整，直接拖拽更简单。

使用动态规则

适合你已经给 Mod 做了标签、类型或分组整理，并希望批量影响排序。

例如：

- 带“语言包”标签的 Mod 靠后。
- 某个作者的一系列框架靠前。
- 某类视觉替换靠后。

动态规则影响范围更大，保存前建议先确认筛选条件没有写得太宽。

常见问题

自动排序是不是越多规则越好？

不是。规则太多、来源冲突或范围过宽，都可能让结果难以理解。优先保留真正有用、能解释清楚的规则。

社区规则和用户规则冲突怎么办？

如果你明确知道自己的环境需要某种顺序，用用户规则表达。用户规则更适合承载你的最终选择。

修改规则后为什么列表没变化？

通常需要重新自动排序。规则保存后只是改变判定依据，不一定立即重排当前列表。

忽略警告安全吗？

只有在你理解原因并确认对当前环境无影响时才忽略。不要用忽略来掩盖红色错误。

注意事项

- 自动排序前建议先扫描，保证 Mod 信息是最新的。
- 排序后必须保存才会写入加载顺序。
- 长期有效的判断写成用户规则，临时测试直接手动拖拽。
- 动态规则范围要收窄，避免误伤大量 Mod。

备份与导入导出说明

这份文档说明 RimCrow 里的加载顺序备份、外部列表导入导出、分享码、软件数据包和模组实体包。它适合需要恢复列表、迁移配置或分享整合包的用户。

先区分两类东西：

- 加载顺序：当前环境启用了哪些 Mod，以及它们的顺序。
- 软件数据或模组文件：RimCrow 设置、规则、环境数据，或者真实 Mod 文件。

加载顺序备份

加载顺序备份用于保存当前环境的启用列表和顺序。它不等于完整备份 Mod 文件。

常见用途：

- 大改排序前留一份回退点。
- 自动排序前后对比。
- 测试不同 Mod 组合。
- 从旧环境恢复某个顺序。

备份页会显示几类记录：

- 临时导入：刚从外部文件或分享码导入，还没有正式保存为备份。
- 最新备份：最近一次相关备份。
- 今日动态：今天产生的备份。
- 历史归档：较早的备份。
- 手动备份：用户另存或整理过的备份。

导入加载顺序

导入用于读取外部列表，并放到对比视图里。导入后不要急着覆盖当前列表，先看差异和缺失项。

支持的常见格式包括：

- `ModsConfig.xml`
- `ModList.xml`
- `.rml`
- RimPy XML
- RimSort JSON
- 文本列表
- Workshop ID 列表
- RimWorld 存档 `.rws`
- RimCrow 分享码

导入后可能出现：

- 当前环境已有的 Mod。
- 当前环境缺失的 Mod。
- 只有包名、暂时找不到来源的项。
- 带工坊 ID，可以尝试补订阅或补下载的项。

导出加载顺序

导出用于把当前启用列表保存成外部文件，方便分享或迁移。

常见格式：

格式	适合场景
<code>ModsConfig.xml</code>	给 RimWorld 或其它工具读取，仅表达包名顺序
<code>ModList.xml</code>	分享列表和工坊 ID 信息
<code>RML</code>	更接近 RimWorld 原生列表格式，适合兼容部分外部流程
分享码	快速复制给别人导入

如果你要让别人尽量补齐 Mod，优先选择包含工坊 ID 或来源信息的格式。

加载备份前要确认什么

加载备份会把当前启用列表恢复到备份状态。当前未保存的更改会丢失。

加载前建议检查：

- 这个备份属于哪个环境。
- 是否有缺失项。
- 是否和当前游戏版本匹配。
- 是否会覆盖你刚手动调整但还没保存的顺序。

如果只是想看差异，先选中备份查看，不要马上加载。

跨环境查看备份

备份页可以切换其它环境的备份。这样你可以查看旧环境或测试环境的加载顺序。

注意：

- 查看不等于切换当前环境。
- 加载到当前环境前要确认缺失项和路径差异。
- 不同环境可能使用不同游戏版本和用户数据目录。

分享码

分享码适合快速把当前启用序列发给别人。

适合：

- 临时分享列表。
- 让别人导入后检查缺失项。
- 不想传文件时使用。

注意：

- 分享码不包含真实 Mod 文件。
- 对方仍需要拥有对应 Mod 来源。

- 分享码里的来源信息越完整，补齐越方便。

推荐清单导出

推荐清单导出适合把整合包整理成更适合阅读和传播的说明材料。

常见格式：

- 文本：简单复制和粘贴。
- Markdown：适合论坛、仓库和文档。
- DOCX / PDF：适合正式分享。
- 图片：适合社群传播。
- 动图：适合展示加载顺序或列表效果。

推荐清单偏“展示和说明”，不等于完整数据备份。

软件数据导入导出

软件数据包用于迁移 RimCrow 自己的数据。

常见内容：

- 设置。
- 规则。
- 提示词。
- 环境数据。
- 其它管理器配置。

适合：

- 换电脑。
- 重装软件。
- 备份自己的配置。
- 在多台机器间同步管理器设置。

导入前建议先保留当前数据备份，避免覆盖后难以回退。

模组实体包导入导出

模组实体包会处理真实 Mod 文件，比加载顺序导出更重。

适合：

- 打包一批本地 Mod 给另一台机器。
- 迁移没有稳定下载来源的 Mod。
- 保存某个整合包当前文件状态。

导出前要注意：

- 文件体积可能很大。
- 工坊 Mod 可能由 Steam 管理，直接复制后更新口径会变化。
- 需要确认磁盘空间。
- 导入时要处理路径、冲突和覆盖风险。

如果只是分享列表，不要用实体包；用推荐清单、分享码或加载顺序导出更轻。

推荐流程

大改排序前

1. 先保存当前列表。
2. 保留一份手动备份。
3. 再自动排序或批量调整。
4. 对比差异。
5. 确认无问题后保存。

导入别人列表

1. 导入文件或分享码。
2. 查看差异。
3. 检查缺失项。
4. 补订阅或补下载。
5. 再决定是否加载到当前环境。

6. 保存。

换电脑

1. 导出软件数据包。
2. 需要真实 Mod 文件时，再导出模组实体包。
3. 新电脑先配置游戏路径和用户数据目录。
4. 导入数据。
5. 扫描。
6. 检查缺失和路径问题。

常见问题

备份会保存 Mod 文件吗？

普通加载顺序备份不会保存真实 Mod 文件。它主要保存启用顺序和相关列表信息。

导入后为什么有缺失项？

导入文件里有某些包名，但当前环境没有找到对应 Mod 文件。可以补订阅、补下载，或手动移除缺失项。

`ModsConfig.xml` 和推荐清单有什么区别？

`ModsConfig.xml` 更偏游戏读取；推荐清单更偏给人看和分享。

软件数据包和模组整合包怎么选？

只迁移 RimCrow 设置、规则和环境数据，用软件数据包。要带真实 Mod 文件，用模组整合包。

注意事项

- 加载备份会覆盖当前启用列表，操作前先确认。

- 导入外部列表后先看差异，不要立刻覆盖。
- 分享码和列表文件不等于 Mod 文件。
- 导出实体包前检查磁盘空间和来源权限。
- 数据异常时先保留现有目录和备份，不要直接删除数据库。

贴图优化说明

这份说明面向普通玩家，解释 RimCrow 贴图优化里常见选项的作用、适合什么情况、可能带来的效果和风险。

先说结论

如果你只是想稳妥改善大型模组列表的贴图读取压力，可以先用：

- 生成范围：只处理新增贴图（增量）或处理全部贴图（覆盖）
- 输出格式：DDS
- 缩放比例：50% 或 60%
- 最小清晰度：128
- 生成 Mipmap：开启

如果你很在意画质，先用 DDS + 不缩放；如果你显存紧张或模组特别多，再逐步降低到 75%、60%、50%。不建议一开始就用 20% 或 25%，除非你能接受部分贴图明显变糊。

贴图优化到底做了什么

RimWorld 模组里常见贴图是 PNG。PNG 适合存储，文件通常比较小，但游戏读取时需要解码，再交给显卡使用。贴图优化会提前把 PNG 生成 DDS 或 DDS.ZSTD，让游戏更容易读取，必要时还可以先缩小图片尺寸。

简单理解：

- PNG 是原图，磁盘上通常省空间，但运行时需要解码。
- DDS 更接近显卡直接使用的贴图格式，适合游戏读取。
- DDS.ZSTD 是把 DDS 再压缩一层，主要用来省磁盘空间。
- 缩放会降低贴图分辨率，主要用来减少显存占用。

贴图优化不会修改原 PNG，而是在旁边生成新的输出文件。这样可以保留原图，也方便以后清理或重新生成。

各种方案有什么区别

方案	做了什么	主要改善	主要代价	适合情况
不处理 PNG	保持原样	无额外文件	仍由游戏读取和解码 PNG	模组少、没有性能问题
原尺寸 DDS	PNG 转 DDS，不缩放	减少运行时解码压力，可能降低贴图卡顿	DDS 通常比 PNG 更占磁盘	想稳妥优化，又不想损失画质
原尺寸 ZSTD	先生成 DDS，再压缩成 <code>.dds.zstd</code>	比 DDS 更省磁盘	需要 Image Opt，读取前仍要解压	磁盘空间紧张，并且使用 Image Opt
缩放 DDS	先缩小图片，再生成 DDS	降低显存占用和贴图体积	画质会下降	大型模组列表、显存压力较大
缩放 ZSTD	缩放后生成 DDS.ZSTD	同时省显存和磁盘	需要 Image Opt，生成更慢	模组很多、显存和磁盘都紧张

游戏原生压缩画质要不要开

RimWorld 自带一个贴图压缩设置，配置项名字是 `压缩画质` (`textureCompression`)。这个选项和 RimCrow 生成 DDS、ZSTD、缩放贴图不是同一个东西。

简单理解：

- RimCrow 的贴图优化是“提前把文件准备好”。
- 游戏原生贴图压缩是“游戏运行时再尽量把贴图压小”。

它们不是二选一。即使你已经生成了 DDS，游戏仍然可能在运行时把一些贴图打包成图集，再根据这个设置压缩图集。所以在大型模组列表里，原生贴图压缩仍然可能明显影响显存占用。

建议怎么选

大多数用户建议保持开启，尤其是这些情况：

- 模组很多。

- 显存不太宽裕。
- 进游戏后容易卡顿、贴图加载慢或接近爆显存。
- 使用了缩放 DDS 或缩放 ZSTD，希望尽量省显存。

可以尝试关闭的情况：

- 显存很充足。
- 你更在意画质细节。
- 想专门对比加载时间。
- 怀疑某些贴图因为压缩出现边缘、渐变或透明区域异常。

开启的好处

- 通常能降低运行时显存占用。
- 大型模组环境下更不容易爆显存。
- 对游戏运行时生成的贴图图集也有作用。
- 可以和 DDS、ZSTD、缩放方案一起使用。

可能的代价

- 启动时可能需要多做压缩工作，所以加载时间不一定更短，有时还会更慢。
- 压缩是有损的，细节、渐变、透明边缘可能略差。
- 改动这个选项后需要重启游戏才会完整生效。

如果不确定，就保持开启。只有在显存充足、追求画质，或者做专门测试时，再尝试关闭。

生成范围怎么选

选项	含义	适合情况
处理全部贴图（覆盖）	重新生成当前范围内所有输出	改了输出格式、缩放比例、Mipmap 后，需要统一刷新
只处理新增贴图（增量）	已存在的输出不动，只补缺失	新增模组后快速补齐
只处理可缩放图片（覆盖）	主要覆盖生成需要缩放的图片	做缩放优化时减少无意义重复生成

如果你不确定参数是否已经统一，先用覆盖生成更直观；日常新增少量模组时，用增量生成更省时间。

DDS

DDS 是贴图优化里最常用、最稳妥的输出格式。RimWorld 1.6 起可以原生读取 DDS；更早版本通常需要 Graphics Settings+ 之类的模组支持。

主要优势：

- 游戏读取贴图时更直接，减少 PNG 解码压力。
- 大型模组环境下，可能减少贴图相关卡顿和爆显存风险。
- 不缩放时一般不会损失画质。

可能风险：

- DDS 文件通常比 PNG 更占磁盘空间。
- 首次生成会耗时，模组越多、贴图越多，耗时越长。
- 如果旧版本游戏或缺少读取 DDS 的支持，游戏可能继续使用 PNG。

ZSTD

ZSTD 输出文件是 `.dds.zstd`。它不是直接替代 DDS 的新贴图格式，而是把 DDS 再压缩一层，主要目标是节省磁盘空间。

主要优势：

- 比普通 DDS 更省磁盘。
- 配合缩放时，可以同时减少磁盘占用和显存压力。

可能风险：

- 需要 Image Opt 才能被游戏读取。
- 如果没有安装或没有启用 Image Opt，仍然可以生成文件，但游戏可能不会使用这些文件。
- 读取时需要额外解压，实际加载表现会受机器、磁盘和 Image Opt 行为影响。

缩放比例

缩放比例决定图片会被压到原来的多大。比例越低，越省显存，但画质越容易下降。

常见选择：

- 不缩放：最稳，不损失清晰度，主要靠 DDS 改善读取。
- 80% / 75%：轻度优化，画质影响较小。
- 60% / 50%：比较均衡，适合大多数大型模组环境。
- 40%：开始偏激进，适合显存压力明显的情况。
- 25% / 20%：非常激进，省资源明显，但细节会变糊。

建议从 60% 或 50% 开始。如果进入游戏后觉得某些贴图太糊，再提高比例重新生成。

最小清晰度

最小清晰度是防止图片被压得太小的保护线。

例如设置为 128 时，缩放会尽量避免把图片最短边压到 128 以下。这样可以保护小图标、小装饰、小 UI 贴图，避免它们被压得看不清。

简单理解：

- 缩放比例决定“想压多少”。
- 最小清晰度决定“最多压到哪里停”。

如果你不确定，保持 128 通常比较稳。

自动回退

有些图片不适合按当前比例缩放，比如缩完会低于最小清晰度。RimCrow 会自动回退到更稳妥的比例，必要时保留原尺寸。

自动回退不是错误。它表示这张图如果强行按当前比例处理，可能太小或不合适。

在结果里看到“回退 50%”“回退 75%”时，意思是这些图片没有使用你当前选择的最低比例，而是改用了更安全的比例。

不缩放、遮罩贴图和超范围

结果里的“不缩放”通常包含三类：

- 不缩放：当前设置或图片条件下保留原尺寸。
- 遮罩贴图不缩放：*_m.png 这类遮罩贴图会生成原尺寸 DDS，但不参与缩放。
- 超范围不缩放：图片尺寸太小或太大，不适合当前缩放规则，保留原尺寸。

遮罩贴图更像是给游戏或渲染使用的数据图，不完全等同普通彩色图片。缩放它们可能造成边缘、通道或显示效果异常，所以更稳妥的策略是生成原尺寸 DDS。

Mipmap

Mipmap 是给远距离显示准备的一组缩小版本。

开启后：

- 远距离看贴图更平滑。
- 闪烁和锯齿感可能更少。
- 文件体积和生成时间会增加一些。

关闭后：

- 输出可能更小，生成可能更快。
- 远距离显示可能更闪，画面稳定性可能下降。

一般建议开启。

清理已生成贴图

清理功能用于删除已经生成的 DDS 或 DDS.ZSTD。常见用途：

- 换了输出格式后，清理旧格式。
- 换了缩放策略后，删除旧输出再重新生成。
- 卸载模组后，清理残留输出。
- 删除没有对应源图的孤立输出。

清理前建议确认目标范围和输出格式，避免删掉你还想保留的优化结果。

加载时间为什么不一定明显变短

贴图优化可能减少贴图读取、解码和上传显卡的压力，但游戏启动时间不只由贴图决定。

加载时间还会受到这些因素影响：

- 系统文件缓存是否命中。
- 磁盘读写和后台程序占用。
- 模组 XML、Defs、PatchOperation 解析。
- C# 模组初始化。
- 音频、预览图、Shader 等其它资源。
- 首次启动和连续启动状态不同。

所以加载时间可以参考，但不适合作为唯一判断标准。更值得关注的是：

- 是否减少进入游戏后的贴图卡顿。
- 是否降低爆显存风险。
- 大型模组列表是否更稳定。
- 磁盘占用是否还能接受。

显存预估为什么和实际不完全一样

RimCrow 里的显存预估用于帮助比较不同方案，不等于游戏进程里看到的实际占用。

原因包括：

- 游戏和显卡驱动会有自己的资源管理和缓存策略。
- Unity 可能保留、复用或延迟释放资源。
- 显存统计工具显示的口径不同。
- 贴图之外还有模型、Shader、RenderTexture、UI、音频缓冲等内容。
- Mipmap、压缩格式、透明通道和实际上传格式都会影响占用。

因此，预估值更适合用来看趋势：比如缩放后理论上应该更省显存；不适合用来和任务管理器里的数字一一对应。

推荐用法

只想稳妥优化

- 输出格式：DDS
- 缩放比例：不缩放
- Mipmap：开启

适合不想损失画质，只想让游戏更容易读取贴图的用户。

大型模组列表，想平衡画质和性能

- 输出格式：DDS
- 缩放比例：50% 或 60%
- 最小清晰度：128
- Mipmap：开启

这是更适合多数大型模组环境的方案。

显存很紧张

- 输出格式：DDS
- 缩放比例：40%、25% 或 20%
- 最小清晰度：128 或更高

这种方案更省资源，但画质风险明显增加。建议先备份或先对部分模组测试。

磁盘空间紧张

- 输出格式：ZSTD
- 确认 Image Opt 已安装并启用
- 缩放比例按画质需求选择

ZSTD 主要解决磁盘占用，不要把它理解成一定更快。

常见问题

生成 DDS 后可以删除 PNG 吗？

不建议。PNG 是原始贴图，也是以后重新生成、回退或排查问题的基础。贴图优化输出是附加文件，不应替代原图管理。

画面变糊怎么办？

提高缩放比例，例如从 50% 改到 75% 或不缩放，然后覆盖重新生成。也可以提高最小清晰度，保护小图。

ZSTD 生成了但游戏没用怎么办？

检查 Image Opt 是否安装并启用。没有 Image Opt 时，DDS.ZSTD 可能只是占在磁盘上，游戏不会读取。

为什么有些图片没有缩放？

可能是遮罩贴图、尺寸超出处理范围，或者缩放后会低于最小清晰度。RimCrow 会尽量避免强行处理这类图片。

我应该什么时候重新生成？

这些情况建议覆盖生成：

- 改了输出格式。
- 改了缩放比例。
- 改了 Mipmap 设置。
- 旧输出统计和当前策略明显不一致。

日常只是新增少量模组，可以用增量生成。

注意事项

- 首次生成可能很慢，贴图越多耗时越长。
- 生成前最好关闭游戏，避免文件占用或结果不一致。
- 大型模组环境建议先从保守设置开始，再逐步降低比例。
- ZSTD 需要 Image Opt; 旧版本游戏读取 DDS 可能需要 Graphics Settings+。
- 贴图优化主要影响贴图读取、磁盘和显存，不会解决所有加载慢或卡顿问题。

日志排错与 AI 诊断说明

这份文档说明 RimCrow 如何查看游戏日志、定位报错，并用 AI 辅助分析问题。它适合遇到红字、启动失败、坏档、卡加载或不确定哪个 Mod 出问题的用户。

先记住：AI 诊断是辅助判断，不是自动修复。真正改动列表前，仍要看日志、Mod 来源和加载顺序。

什么时候看日志

这些情况建议先看游戏日志：

- 游戏启动失败。
- 加载到一半卡住。
- 进入存档后红字很多。
- 某个功能界面打不开。
- 保存、生成地图、战斗或事件时反复报错。
- 加了新 Mod 后出现异常。

如果是 RimCrow 软件本身报错，可以切到系统日志。

日志中心里有什么

日志中心主要有两个来源：

- 游戏日志：RimWorld 的运行日志，适合排查 Mod 和游戏错误。
- 系统日志：RimCrow 自己的运行日志，适合排查软件操作、路径、下载和接口问题。

普通玩家优先看游戏日志。

基本排错流程

1. 先复现问题

尽量明确问题发生在什么时候：

- 启动阶段。
- 加载存档。
- 进入地图后。
- 点击某个按钮。
- 某个事件触发后。

复现后再打开日志，日志更容易定位到相关错误。

2. 找到关键错误

优先关注：

- Error
- Exception
- Could not load
- Failed
- PatchOperation
- 反复出现的同一段报错。

不要只看最后一行。真正原因有时在第一段红字或重复错误的起点。

3. 结合 Mod 名和包名

日志里可能出现：

- Mod 名称。
- PackageId。
- DLL 名称。
- XML 路径。
- 贴图或 Def 路径。

把这些关键词拿回主列表搜索，通常能定位到可疑 Mod。

4. 调整后再测试

处理方式可能包括：

- 补齐依赖。
- 调整加载顺序。
- 禁用可疑 Mod。
- 更新或回退 Mod。
- 加兼容补丁。
- 清理残留配置。

每次只改少量内容，方便确认是哪一步生效。

一键分析

一键分析适合日志很长、错误很多、你不知道从哪里开始看的情况。

它会先压缩和整理全局错误，再交给 AI 分析。适合：

- 红字太多。
- 同类错误重复刷屏。
- 不确定哪个错误最关键。
- 想要一个优先级列表。

分析结果通常会给出：

- 最可疑的错误来源。
- 可能相关的 Mod。
- 推荐处理顺序。
- 需要继续确认的信息。

手动选择日志分析

如果你已经知道哪一段日志最相关，可以手动勾选日志，再打开 AI 助手。

适合：

- 只想分析某一段报错。
- 想围绕某个 Mod 继续追问。
- 想减少无关日志干扰。

选中日志后，RimCrow 会估算 Token 占用。接近模型上限时，建议减少选择范围或改用一键分析。

AI 助手怎么问更有效

不要只问“怎么修”。更推荐这样问：

- “按优先级列出最可疑的 3 个 Mod。”
- “先判断这是缺依赖、排序问题还是兼容冲突。”
- “给我一个最小测试步骤。”
- “哪些报错可以暂时忽略，哪些必须先处理？”
- “如果我要二分排查，第一轮应该禁用哪些 Mod？”

问题越具体，结果越容易执行。

工具权限

AI 侧栏可能允许读取日志上下文、模组资料、排序规则等信息。

复杂问题建议保持默认开启，因为只看一段日志往往不够；AI 需要结合当前列表、规则和 Mod 信息才能判断。

如果你只想让 AI 看你手动选择的日志，可以减少工具权限。

辅助工具模组

日志中心里有“使用辅助工具模组”开关。开启后，RimCrow 可以在保存或自动排序时自动启用辅助工具模组，为软件提供更详细的游戏日志信息。

普通用户可以保持默认。只有在你明确不想加入任何工具模组时，再关闭。

文件内容搜索

文件内容搜索适合查：

- 某个 Def 名称来自哪个 Mod。
- 日志里提到的 XML 路径。
- 某个关键词在哪些 Mod 文件里出现。
- 怀疑某个补丁或翻译文件造成问题。

搜索结果可以帮助你从日志里的技术词定位到具体 Mod 文件。

常见排错场景

启动阶段报错

优先检查：

- 缺少前置。
- 加载顺序错误。
- DLL 或版本不兼容。
- 游戏版本和 Mod 版本不匹配。

进入存档后报错

优先检查：

- 存档里已经存在的物品、派系、建筑或生物是否来自被删除 Mod。
- 最近是否移除了大 Mod。
- 是否更新了改变存档结构的 Mod。

大量 PatchOperation 报错

通常和 XML 补丁有关。可能是：

- 被补丁引用的目标不存在。
- 前置 Mod 没启用。

- Mod 版本变化后路径变了。
- 补丁 Mod 排序不合适。

某个贴图或资源加载失败

优先检查：

- 文件是否缺失。
- Mod 是否下载完整。
- 路径大小写或特殊字符。
- 是否刚做过贴图优化或清理。

贴图优化相关说明见 [贴图优化说明](#)。

隐私和安全

日志和 AI 请求可能包含：

- 本机路径。
- 用户名。
- Mod 列表。
- 存档名。
- 部分配置或报错上下文。

分享给别人或发送给 AI 前，建议检查是否包含你不想公开的信息。

AI API Key、代理账号等敏感字段应通过设置保存，不要写进日志或聊天内容。

常见问题

AI 说的可疑 Mod 一定对吗？

不一定。AI 是根据日志和上下文推断。把它当成排查优先级建议，不要当成最终结论。

日志太长发不出去怎么办？

用一键分析，或只选择最相关的错误段。不要整份日志无脑发给 AI。

系统日志和游戏日志该看哪个？

游戏红字、坏档、Mod 报错看游戏日志。RimCrow 操作失败、下载失败、路径失败看系统日志。

改完后怎么确认修好了？

重新扫描、保存、启动游戏，复现同一个操作。确认原来的关键错误不再出现，再继续加回其它 Mod 或做下一步调整。

注意事项

- 每次只改少量内容，方便确认原因。
- 不要只根据一条日志就删除大量 Mod。
- AI 诊断结果要结合 Mod 说明和实际测试。
- 分享日志前检查本地路径和个人信息。
- 数据异常时先备份，不要直接删除用户数据目录。