

Dibab

用户手册

在您自己的机器上整理家庭的行政文件

版本 0.9.0-alpha

2026-11-26

译文。以英文版为准。

dibab.davalan.fr

目录

简介 **3**

Dibab 做什么 . . . 3

它不做什么 . . . 3

为家庭打造的工具 . . . 3

本手册的结构 . . . 3

约定 . . . 4

安装 **5**

您需要什么 . . . 5

按角色划分的模型 . . . 5

最终目录树 . . . 5

用 Debian 软件包安装 . . . 6

在 Fedora、openSUSE Leap 和 Arch 上安装 . . . 6

从源代码安装 . . . 7

运行测试（可选） . . . 8

配置的查找位置 . . . 8

配置 **9**

必须填写的键 . . . 9

`destinations`：各领域的保险库 . . . 9

`destination`、`arborescence`、`executer_sur` . . . 10

`suivi`：状态、日志与报告 . . . 10

`ollama`：模型 . . . 11

`extraction` 与 `classification` . . . 11

`categories`、`personnes`、`emetteurs_connus`、`pieces_communes` . . . 12

`nommage`：归档文件的名称 . . . 13

`extensions_cibles` 与 `scan` . . . 13

`sources_locales` . . . 14

`ssh`、`sources_distantes`、`imap`、`oauth2`、`sources_imap` . . . 14

`mesure` 与 `inventaire` . . . 14

首次处理 **15**

模拟：`--dry-run` . . . 15

不用模型检查归档路径 . . . 17

归档：`--execute` . . . 18

撤销：`--rollback` . . . 20

退出码 . . . 20

保险库与加密 **21**

什么是保险库 . . . 21
挂载确认 . . . 21
gocryptfs 示例 . . . 22
状态存放在哪里 . . . 22
磁盘空间 . . . 23

通过 ssh 访问的来源 **24**
远程机器必须提供什么 . . . 24
声明机器 . . . 24
首次运行：先列出，再取回 . . . 25
再次运行 . . . 27
连接失败时 . . . 27

IMAP 邮箱与 Google 账户 **29**
向服务器请求什么 . . . 29
声明邮箱 . . . 29
密码 . . . 30
首次运行：先列出，再存放 . . . 30
连接失败时 . . . 32
使用 OAuth2 的 Google 账户 . . . 33

清单与轮转 **35**
读取保险库中已有的内容：`--inventaire` . . . 35
控制日志大小：轮转 . . . 36

诊断 **37**
状态 . . . 37
按顺序列出的检查 . . . 38
阅读失败结果 . . . 39
全新安装 . . . 40
使用另一种语言 . . . 40

自主性与局限 **41**
工具保证什么 . . . 41
尊重家庭成员 . . . 41
局限 . . . 41
报告问题 . . . 42

简介

Dibab（布列塔尼语意为“选择”）收集、分析并归档一个家庭的个人行政文件：发票、信件、合同、公共机构的通知。它把每份文件移入规范化的目录树，使用说明日期、发件机构和类别的文件名，并为所做的一切记录日志，使任何归档都可以撤销。本手册描述版本 0.9.0-alpha。

它以自主模式运行。任何数据都不离开您的本地网络：没有第三方服务，没有远程模型，也没有云。语言模型是运行在您自己机器上的 Ollama 提供的模型。

Dibab 做什么

- 它扫描本地目录、通过 `ssh` 访问的远程机器以及通过 IMAP 访问的邮箱。远程文件经复制并用指纹校验，原件保持原样。邮箱只读取而不修改：只取回附件，不把任何邮件标为已读。
- 它提取每个文件的文本（PDF 的文本层，否则由视觉模型识读页面图像），剔除非行政文件，然后在多个模型达成一致后，按类别和领域（个人或业务）归档其余文件。
- 它重命名每份文件并移入相应领域的保险库，写入带时间戳的日志，并可用 `--rollback` 撤销整次运行。

它不做什么

- 它不移动没有把握的文件。有疑问就放入名为 `待核实/` 的文件夹，绝不冒险归档。
- 它从不删除文件。它先复制、校验，只有在确认副本完全相同后才移除原件。
- 它不修改邮箱或远程机器。
- 它不在后台运行。由您手动启动，默认模式是模拟，即 `--dry-run`。
- 它不挂载、不加密、不同步保险库。这些由您负责，第 5 章说明工具如何检查保险库。

为家庭打造的工具

Dibab 为家庭的行政事务而设计，由家庭自己使用。只有在当事人明示或默示同意的情况下，一台机器、一个目录或一个邮件账户才纳入范围；有疑问时就不纳入。软件无法保证做到这种尊重：实例的管理员对此作出承诺。

日志记录文件名和发件机构，这些已经透露很多信息（一封医疗机构的来信就足够）。因此它应当放在保险库内，而不是旁边：把 `suivi.*` 各键指向保险库内部，否则它会以明文写在 `~/.local/state/dibab/` 下。

本手册的结构

1. 简介（本章）。
2. 安装：需要安装什么、各部分放在哪里，以及首次检查。
3. 配置：配置文件，逐键说明。
4. 首次处理：模拟、阅读报告、实际归档、撤销。
5. 保险库与加密：如何确认保险库已挂载，以及状态保存在哪里。
6. 通过 ssh 访问的来源：远程机器。
7. IMAP 邮箱与 Google 账户。

8. 清单与轮转：读取保险库中已有的内容，并控制日志大小。
9. 诊断：`--diagnose`，逐一说明各状态。
10. 自主性与局限：工具保证什么，不保证什么。

约定

命令按照在终端中输入的形式给出，由第 2 章所述安装的用户执行。命令名是 `dibab`。命令后面的文本是它的实际输出，通过在由虚构文件组成的试验区中重放命令得到；这些输出中的模型名称（`essai-ocr`、`essai-tri`、`essai-classification`）是试验区的替身模型，您使用的是配置中写明的模型。本手册中的所有文件、人员、发件机构和公司均为虚构。

消息语言跟随您的会话：Dibab 读取 `LC_ALL`、`LC_MESSAGES` 和 `LANG` 中第一个非空的变量，并相应使用英文、法文、德文、西班牙文或简体中文（其他值使用英文）。选项名称、`OK` 和 `ÉCHEC` 等状态标记、标识符、日志和计划文件从不随语言变化，因此用一种语言写下的日志可以在任何其他语言下用 `--rollback` 回放。本手册中的输出为英文。

安装

您需要什么

- 装有 Python 3.10 或更高版本的 Linux 系统。Dibab 只使用 Python 标准库，因此无需为 Python 安装其他内容。
- `findmnt`（软件包 `util-linux`），用于确认保险库已挂载。
- `pdftotext` 和 `pdftoppm`（软件包 `poppler-utils`），用于读取 PDF 文件。
- `ssh`（软件包 `openssh-client`），仅当您要扫描远程机器时需要，并使用无需密码即可打开会话的密钥。
- 已安装并正在运行的 Ollama，每个角色至少一个模型（见下文）。Dibab 从不安装 Ollama，也从不下载模型。
- 个人和业务两个领域各一个目标目录，最好位于加密卷上（第 5 章）。
- `git`，用于获取源代码。

在 Debian 或 Ubuntu 上，一条命令即可安装这些软件包：

```
sudo apt install python3 git util-linux poppler-utils openssh-client
```

按角色划分的模型

三个角色各使用一个模型。角色 `ocr` 识读扫描页面，`tri_rapide`（“快速分拣”）剔除非行政文件，`classification` 进行归档。`classification` 角色的模型进行表决：只有当它们的一致度达到 `ollama.seuil_confiance` 时，文件才会归档（第 3 章）。两个可选角色 `tri_decision` 和 `objet` 让一个 `decision` 模型一次完成分拣，并让一个小模型给出主题；没有它们时由 `tri_rapide` 分拣（第 3 章）。

```
ollama pull glm-ocr
ollama pull mistral-nemo:12b-instruct-2407-q6_K
ollama pull gemma4
ollama pull gemma3:4b
ollama pull qwen2.5
ollama pull granite4.2:8b-Q5_K_M
```

这些是参考实例使用的模型；其他模型同样可以，只要在 `ollama.roles` 中写明。用 `ollama list` 列出已安装的模型。

最终目录树

~ 是您的主目录。请按您的安装调整路径。

```
~/applications/dibab/              克隆的仓库
├── dibab.py                       入口文件
├── modules/                           每项职责一个文件
│   └── langues/                       消息词典： en, fr, de, es, zh
├── docs/                              手册页和 README 译文
├── tests/                             测试
```

```
└── config.exemple.json                 示例配置
~/.local/bin/dibab                  指向 dibab.py 的链接：即命令
~/.config/dibab/config.json         您的配置，从不纳入版本管理
~/.local/state/dibab/               状态、日志和运行报告（默认位置）
~/Classement/Personnel/                 个人领域的保险库
~/Classement/Professionnel/             业务领域的保险库
```

用 Debian 软件包安装

在 Debian 12 和 13 以及 Ubuntu 22.04 和 24.04 上，一个文件即可安装程序及其依赖（`python3`、`poppler-utils`、`util-linux`、`openssh-client`）。它不包含 Ollama、模型和配置。从 dibab.davalan.fr 的下载页面下载 `dibab_0.9.0-alpha_all.deb`，然后在该文件所在文件夹中执行：

```
sudo apt install ./dibab_0.9.0-alpha_all.deb
```

以当前会话的语言写入配置，然后按下文从源代码安装的第 4 步和第 5 步操作（每个领域的保险库、配置中的键）：

```
dibab --initialiser
```

`--initialiser` 以当前会话的语言写入配置：核查文件夹、其前缀和重复文件夹采用该语言的名称。它在权限为 `700` 的文件夹中以 `600` 权限创建文件，从不改写已经存在的文件；之后用另一种语言运行也不会重命名它们。

检查安装，不写入任何内容：

```
dibab --version
dibab --diagnose
```

软件包把命令放在 `/usr/bin/dibab`，程序放在 `/usr/lib/dibab/`，README、示例配置和手册页（`man dibab`，五种语言）放在 `/usr/share/` 下。卸载会保留您的数据：`sudo apt remove dibab` 和 `sudo apt purge dibab` 只删除软件包的文件，从不触碰您的配置、状态、日志和文件；清除时会列出它保留的内容。

验证程度：安装和诊断已在容器中于 Debian 12 和 13 以及 Ubuntu 22.04 和 24.04 上验证（安装、`--version`、`--diagnose`、无配置时拒绝写入、保险库未挂载时拒绝写入、一次模拟归档、卸载和清除）。容器无法验证加密保险库的挂载，也无法验证使用您自己的 Ollama 和文件的运行。

在 Fedora、openSUSE Leap 和 Arch 上安装

在 Fedora 44、openSUSE Leap 16.0 和 Arch Linux 上，同样只需一个文件即可安装程序及其依赖。从 dibab.davalan.fr 的下载页面下载适用于您系统的文件，然后在该文件所在文件夹中执行您系统对应的那一行：

```
sudo dnf install ./dibab-*.fc44.noarch.rpm
sudo zypper install --allow-unsigned-rpm ./dibab-*.leap160.noarch.rpm
sudo pacman -U ./dibab-*-any.pkg.tar.zst
```

这些软件包没有签名。其余步骤与 Debian 软件包相同：`dibab --initialiser`，然后是 `dibab --version` 和 `dibab --diagnose`。卸载软件包（`sudo dnf remove dibab`、`sudo zypper remove dibab`、`sudo pacman -R dibab`）从不触碰您的配置、状态、日志和文件。

验证程度：安装和诊断已在容器中于 Fedora 44、openSUSE Leap 16.0 和 Arch Linux 上验证（安装、`--version`、`--diagnose`、两种拒绝写入的情况、一次模拟归档、覆盖损坏的安装进行重装、卸载）。容器无法验证加密保险库的挂载，也无法验证使用您自己的 Ollama 和文件的运行。

从源代码安装

1. 获取源代码：

```
mkdir -p ~/applications
git clone https://github.com/RonanDavalan/dibab.git ~/applications/dibab
```

2. 使命令可用。入口文件是可执行的；在 `~/.local/bin` 中建立一个链接即可让它以 `dibab` 为名（在大多数发行版上，该目录已经在登录 shell 的 `PATH` 中）：

```
mkdir -p ~/.local/bin
ln -s ~/applications/dibab/dibab.py ~/.local/bin/dibab
```

3. 以当前会话的语言写入配置（第 3 章说明每个键）：

```
dibab --initialiser
```

`--initialiser` 以当前会话的语言写入配置：核查文件夹、其前缀和重复文件夹采用该语言的名称。它在权限为 `700` 的文件夹中以 `600` 权限创建文件，从不改写已经存在的文件；之后用另一种语言运行也不会重命名它们。

4. 为每个领域创建保险库，并读取其文件系统类型（原因见第 5 章）：

```
mkdir -p ~/Classement/Personnel ~/Classement/Professionnel
findmnt -n -T ~/Classement/Personnel -o FSTYPE
```

```
ext4
```

该命令输出承载目录的卷的类型，例如 `ext4`。对于普通目录，把此类型写入 `destinations.<domain>.systemes_fichiers_acceptes`；或者先打开您的加密卷，写入 `findmnt` 为它报告的类型。

5. 编辑 `~/.config/dibab/config.json`，至少填写第 3 章标为必填的键，然后检查安装，不写入任何内容：

```
dibab --version
```

```
dibab 0.9.0-alpha
```

该命令输出一行后停止。它不读取配置，不触碰任何文件，也不与任何服务通信。

运行测试（可选）

源代码自带测试，需要在虚拟环境中安装 `pytest`。使用 Dibab 并不需要它们；请在更新之后，或怀疑安装有问题时运行：

```
cd ~/applications/dibab && \
  python3 -m venv venv && \
  venv/bin/pip install -r requirements-dev.txt && \
  venv/bin/python -m pytest -q tests
```

最后一行报告通过和跳过的测试数；跳过的是针对正在运行的 Ollama 的试验，测试套件并不要求它。

配置的查找位置

Dibab 按以下顺序读取找到的第一个文件：`--config <path>` 选项、`DIBAB_CONFIG` 环境变量、`$XDG_CONFIG_HOME/dibab/config.json`（该变量为空时为 `~/.config/dibab/config.json`），最后是 `dibab.py` 旁的 `config.json`。不做合并：只读取找到的第一个文件。如果选项或变量指向不存在的文件，程序以退出码 1 停止。它绝不退而使用另一个文件，因为那会使它在您没有指定的配置上工作，也就是在另一个保险库上工作。

配置

Dibab 中没有任何硬编码。每个路径、模型名称、阈值、时限和命名规则都在一个 JSON 文件 `config.json` 中。改变工具的行为就是修改这个文件，而不是代码。源代码附有一个完整且贴近实际的示例 `config.exemple.json`；本章逐块说明。第 2 章说明程序在哪里查找该文件。

以 `~` 或 `$HOME` 书写的路径会被展开。示例中没有绝对路径。任何机密都不写入该文件：邮件密码放在环境变量中，Google 令牌放在权限为 `600` 的单独文件中（第 7 章）。

缺失的键不会被默默地用猜测值代替。程序以退出码 1 停止并指出缺失的键，因此修改之后首先应运行 `dibab --diagnose`（第 9 章）。

必须填写的键

打开 `dibab --initialiser` 写出的文件，然后至少修改以下内容：

键	填写内容
`destinations.personnel.racine`, `destinations.professionnel.racine`	各领域保险库的根目录
`destinations.*.systemes_fichiers_acceptes`	用于确认保险库已挂载的文件系统类型（第 5 章）
`personnes`	家庭成员的名字；用于命名个人领域的文件夹
`sources_locales[].chemin`	要扫描的目录
`ollama.roles`	每个角色按顺序列出首选模型（`config.exemple.json` 把这三个列表留空）
`suivi.fichier_date`, `suivi.journal`, `suivi.sessions`	可选，但建议放在保险库内（第 5 章）

删除 `sources_distantes` 和 `sources_imap` 中您不使用的示例条目：声明了的来源就是会被读取的来源。

`destinations`：各领域的保险库

每个领域一个条目，即 `personnel` 和 `professionnel`。

键	含义
`racine`	该领域保险库的根目录
`systemes_fichiers_acceptes`	用于确认保险库已挂载的文件系统类型；示例期望 `fuse.gocryptfs`
`sous_repertoire_classement`	根目录下归档本领域文件的文件夹（个人领域为 `Administratif`，业务领域为 `Classement`）

键	含义
sous_repertoire_a_verifier	存放 Dibab 没有把握的文件的文件夹（待核实/）

核查文件夹的名称及其前缀在初始化时按会话语言写入：sous_repertoire_a_verifier 设定文件夹，nommage.prefixe_a_verifier 设定前缀。保险库根目录下接收重复文件的文件夹由 destination.repertoire_doublon 设定。

destination、arborescence、executer_sur

键	含义
destination.sous_repertoire_logs	保险库内业务日志的文件夹（logs）；清单计划写在这里
destination.vide_poches_prefixe	“投放文件夹”的前缀（vide-poches-），取回的文件在此等待
destination.repertoire_doublon	保险库根目录下的文件夹，接收指纹已归档过的文件（重复/）
arborescence.personnel	个人文件在其归档文件夹下的子路径：{personne}/{annee} 得到 Alice/2026
arborescence.professionnel	业务文件同理：{annee}/{categorie} 得到 2026/Factures
arborescence.personne_inconnue	人员不在 personnes 中时使用的文件夹（Commun）
executer_sur	允许执行 --execute 的机器名（hostname）；空列表表示任何机器，在其他机器上 --execute 以退出码 4 停止

suivi：状态、日志与报告

键	含义
fichier_date	文件 dernier_scan.json，记录每个来源上次扫描的日期
journal	操作日志 journal.jsonl，每个操作一行 JSON，由 --rollback 读回；它记录路径，从不记录文件文本
sessions	运行报告的文件夹
logs_systeme	为技术日志保留；目前没有任何内容写入它
securite_jours	新扫描从上次扫描前多少天重新开始（1），以免漏掉同时添加的文件
premier_scan_jours	某个来源的首次扫描回溯多久（90 天）
rotation.journal_taille_max_mo	日志超过该大小时进行轮转（20 MB）
rotation.journal_archives_max	保留多少份轮转后的日志（5）
rotation.sessions_max	保留多少个运行文件（100）

四个路径键（fichier_date、journal、sessions、logs_systeme）是可选的。缺省时文件放在 $XDG_STATE_HOME/dibab/（即 ~/.local/state/dibab/）下，名为 dernier_scan.json、journal.jsonl、sessions/ 和 dibab.log，文件权限为 600，文件夹权限为 700。这些默认位置在磁盘上是明文：第 5 章说明如何把它们移入保险库。另有三个文件始终与 fichier_date 放在一起：rapatries.json、imap_deposes.json 和 dibab.verrou。

ollama：模型

键	含义
url	Ollama 的地址（http://localhost:11434）；每次调用都是发往它的普通 HTTP 请求
roles.ocr	识读扫描页面的模型的首选顺序
roles.tri_rapide	剔除非行政文件的模型的首选顺序
roles.classification	进行归档的模型；它们全部处理每份文件并表决
roles.tri_decision	可选：一个 decision 模型（ollama show 会列出该能力）的首选顺序，它按阈值一次完成分拣，代替 tri_rapide
roles.objet	可选：当由决策模型分拣时，给出文件主题的小型生成模型
seuil_confiance	介于 0 和 1 之间；只有当意见一致的归类模型比例达到它时才保留领域（示例中为 0.4）
timeout_secondes	对 Ollama 每个请求的时限（120）
keep_alive_entre_pages	OCR 模型在一份文件的两页之间保持加载的时间（2m）

启动时，Dibab 向 Ollama 查询已安装的模型，对 ocr 和 tri_rapide 保留第一个已安装的首选项；不带标签的名称代表 :latest。对 classification，它保留所有已安装的首选项，并由它们表决。某个角色没有任何已安装模型时，程序在写入任何内容之前以退出码 3 停止，列出现有模型并指出需要修改的行。两个可选角色从不使程序停止：当声明了 tri_decision 但无法使用时（没有已安装的模型、没有 decision 能力、没有阈值、objet 没有模型，或 Ollama 不支持 /v1/systemone），分拣改用 tri_rapide，运行开头会说明原因。

extraction 与 classification

键	含义
extraction.seuil_caracteres_alphanumeriques	字母和数字少于该数量时，PDF 由 OCR 模型识读而不用其文本层（200）
extraction.pages_ocr	OCR 读取的页数（3）
extraction.resolution_ocr	渲染页面的分辨率，单位 dpi（200）
extraction.taille_max_texte	提取的文本截断到该字符数（20000）
extraction.repertoire_temporaire	页面渲染位置；应保持在内存中（/dev/shm）
extraction.timeout_outils_secondes	pdftotext 和 pdftoppm 的时限（60）

键	含义
extraction.consigne_ocr	随每张图像发给 OCR 模型的指令
classification.temperature, classification.num_ctx	随每个请求发送的选项（0 和 8192）；上下文过短会截掉提示的开头
classification.caracteres_max_prompt	发给归类模型的文本量（12000）
classification.caracteres_max_tri	发给分拣模型的文本量（3000）；文件开头说明它是什么
classification.keep_alive_entre_documents	模型在两份文件之间保持加载的时间（2m）
classification.longueur_max_champ, longueur_max_citation	保留的发件机构、收件人和主题的最大长度，以及引述片段的最大长度（80 和 120）
classification.marqueurs_mentions	出现即为弱业务信号的词（TVA、SIRET 等）
classification.seuils_tri_decision	可选：每个决策模型的 non_administratif 和 administratif（介于 0 和 1 之间）以及 caracteres_min；模型给出的非行政概率从前者起判为非行政，低于后者判为行政，两者之间判为不确定，字母和数字少于 caracteres_min 的文本判为不确定。阈值只对一个模型和一个 caracteres_max_tri 有效：任一改变都要重新测量

文件文本一律视为数据，绝不视为指令：它由随机标记界定，模型以封闭的 JSON 作答并由 Dibab 校验，任何无效应答都计为“不确定”。模型永远无法触发任何操作。

categories、personnes、emetteurs_connus、pieces_communes

键	含义
categories	类别的封闭列表（Factures、Aides、Impôts 等）；模型只能回答其中之一或“无”
personnes	家庭成员；模型只能回答其中之一或“无”
emetteurs_connus	按类别列出的已知发件机构，每个为 {"nom": ..., "domaine": "personnel" | "professionnel" | "mixte"}；已知发件机构是领域信号，空列表则不提供该信号
pieces_communes	仅当所有归类模型返回同一类别时，以相同名称和相同指纹复制到各领域保险库的类别（Identité、Domicile、RIB）

示例中的类别和已知发件机构是通用的；请按您所在国家调整。

nommage：归档文件的名称

键	含义
format	归档文件名的模板，包含 {date}、{emetteur}、{categorie}、{personne}、{objet}（分拣返回的主题）、{ext}（{date}_{emetteur}_{categorie}{ext}）
separateur, espaces_vers	分隔符和空格的替换字符（_ 和 -）
date_format	日期格式（%Y-%m-%d）
date_inconnue_prefixe	文件中找不到日期时，所用修改日期的前缀（~）
emetteur_inconnu	发件机构未知时使用的名称（Inconnu）
prefixe_a_verifier	放入核查文件夹的文件的前缀（[待核实]）
format_a_verifier	前缀之后的模板，变量相同，另有 {origine} 即原文件名；空白或未知部分连同分隔符省略，整个名称不超过 180 个字符（{date}_{emetteur}_{objet}_{origine}{ext}）；没有此键时文件保留原名
suffixe_collision	名称已被占用时的后缀，从 2 开始（_{n}）

使用 dibab --initialiser 写出的配置时，一份日期为 2026 年三月 15 日的公共机构通知会命名为 2026-03-15_Organisme-public_Aides.pdf，一封放入核查文件夹的无抬头信件会命名为 [待核实]courrier-de-suivi-de-dossier_courrier-sans-en-tete.pdf。

extensions_cibles 与 scan

键	含义
extensions_cibles	要处理的扩展名（.pdf、.png、.jpg、.jpeg），比较时不区分大小写
scan.recursif_par_defaut	扫描子文件夹（true）；每个来源可以覆盖
scan.exclusions_automatiques	对所有来源都从不扫描的子文件夹（rebut，以及 destination.repertoire_doublon 指定的文件夹）
scan.suivre_symlinks	跟随符号链接（false，使扫描无法离开其范围）
scan.ignorer_dotfiles	忽略隐藏文件和文件夹（true）

sources_locales

每个条目是一个要扫描的目录。

键	含义
id	唯一名称；它是该来源在 dernier_scan.json 中的键
chemin	目录
date_reference	modification（推荐）或 creation；用于筛选近期文件的日期
recursif	对该来源覆盖 scan.recursif_par_defaut
exclusions	仅对该来源跳过的子文件夹名称

某个来源的首次扫描回溯 suivi.premier_scan_jours 天；之后的扫描从上次完整扫描的日期减去 suivi.securite_jours 开始。

ssh、sources_distantes、imap、oauth2、sources_imap

这些区块在使用处说明：ssh 和 sources_distantes 见第 6 章，imap、oauth2 和 sources_imap 见第 7 章。

mesure 与 inventaire

键	含义
inventaire.exclusions	--inventaire 不读取的文件夹名称（第 8 章）
mesure.graine, documents_par_coffre, exclusions, taux_erreur_max	仅由源代码中的测量工具（_outils/mesurer_domaines.py）读取，它对已归档文件抽样并对照保险库检查模型；示例值只是起点

首次处理

首次处理分四个环节：把文件放入来源目录，用 `--dry-run` 模拟，阅读模拟给出的建议，然后用 `--execute` 实际归档。每次归档都可以用 `--rollback` 撤销。始终从模拟开始：它不移动任何内容，不在任何保险库中写入，并准确显示每份文件的去向。

本章使用试验区配置中的来源目录 `~/Classement/Entrant`，其中有四份虚构文件：一份公共机构通知、一张主机托管发票、一份蛋糕食谱和一封没有任何抬头的信件。

```
ls ~/Classement/Entrant
```

```
courrier-sans-en-tete.pdf
facture-hebergement.pdf
notification-organisme.pdf
recette-gateau.pdf
```

模拟：`--dry-run`

`--dry-run` 是默认模式：单独运行 `dibab` 效果相同。它需要 Ollama，因为整个归类过程都会运行，但它不移动任何文件，既不写 `dernier_scan.json`，也不写任何归档文件。

```
dibab --dry-run
```

```
Dibab 0.9.0-alpha — session 2026-11-23T12:09:03
  Mode               : dry-run
  Configuration      : ~/.config/dibab/config.json
  Vault personnel     : ~/Classement/Personnel — mounted
  Vault professionnel : ~/Classement/Professionnel — mounted
  Session journal    : ~/.local/state/dibab/sessions/session_2026-11-23T12:09:03.jsonl
  Model ocr           : essai-ocr:latest
  Model tri_rapide    : essai-tri:latest
  Model classification: essai-classification:latest
  Consensus          : 1 classification model(s), threshold 0.4

Source: entrant (~/Classement/Entrant)
  - notification-organisme.pdf                        (928 B, mtime 2026-11-23)
  - facture-hebergement.pdf                           (922 B, mtime 2026-11-23)
  - recette-gateau.pdf                                (895 B, mtime 2026-11-23)
  - courrier-sans-en-tete.pdf                         (901 B, mtime 2026-11-23)
  Source total: 4 file(s), 3.6 KB

============================================================
Scan total: 4 candidate file(s), 3.6 KB
============================================================
Text extraction (the text is neither shown nor written)
```

```
  - notification-organisme.pdf: text layer, 258 character(s), 1 page(s), 0.02 s
  - facture-hebergement.pdf: text layer, 246 character(s), 1 page(s), 0.02 s
  - recette-gateau.pdf: text layer, 226 character(s), 1 page(s), 0.02 s
  - courrier-sans-en-tete.pdf: text layer, 236 character(s), 1 page(s), 0.02 s

Classification (dry-run, nothing is moved; the text is neither shown nor written)
model 1/1, document 1/3
model 1/1, document 2/3
model 1/1, document 3/3
  - notification-organisme.pdf: triage administrative, domain personnel, agreement 1/1
      subject: notification de paiement d'allocation
      -> vault personnel, classement: ~/Classement/Personnel/Administratif/Alice/2026/2026-03-1
       ↪ 5_Organisme-public_Aides.pdf
  - facture-hebergement.pdf: triage administrative, domain professionnel, agreement 1/1
      subject: facture d'hébergement
      -> vault professionnel, classement: ~/Classement/Professionnel/Classement/2026/Factures/2
       ↪ 026-03-15_Hebergeur-Exemple_Factures.pdf
  - recette-gateau.pdf: triage not administrative, domain not applicable
      subject: recette de gâteau
      -> stays in place (not administrative)
  - courrier-sans-en-tete.pdf: triage administrative, domain incertain, agreement 1/1
      subject: courrier de suivi de dossier
      reason: no recipient quoted and no known issuer
      -> vault personnel, a_verifier: ~/Classement/Personnel/待核实/[待核实]courrier-de-suivi-de-
       ↪ dossier_courrier-sans-en-tete.pdf (not moved)
```

进度行（`model 1/1, document 1/3`）写到错误输出，每个模型、每份文件各一行。首次处理较慢，因为每个归类模型都要阅读每份文件，这些行表明程序并未卡住。

阅读报告

报告按以下顺序分为四部分。

1. 开头：版本和运行时间、模式、读取的配置文件、每个保险库及其已挂载的确认、运行日志的路径、每个角色保留的模型以及共识规则。
2. 扫描：对每个来源，列出候选文件及其大小和日期，然后是合计。文件的扩展名在 `extensions_cibles` 中且日期足够新时才成为候选。
3. 文本提取：对每个候选文件，说明文本的获取方式（`text layer`，或 OCR 及识读它的模型）、字符数和页数以及耗时。文本本身既不显示也不写入。
4. 归类：每份文件一个区块。第一行给出分拣结果（行政、非行政或不确定）、Dibab 保留的领域和一致度，即在被询问的归类模型中返回该领域的数量（`agreement 2/3`：三个模型中有两个）。`subject:` 行用几个词说明文件是什么，依据分拣模型的识读；它绝不是文本摘录。对于送往核查文件夹的文件，`reason:` 行从封闭列表中给出原因。最后一行在 `->` 之后给出目标位置。

实际执行时，最后一行给出运行报告的名称。

目标位置写作 `-> vault <domain>, <zone>: <path>`。区域说明将发生什么：

区域或文字	含义
`classement`	文件以规范化名称归入该保险库的归档区
`a_verifier`	Dibab 没有把握：文件进入核查文件夹 **待核实/**，名称以前缀 **[待核实]** 开头，后接文件给出的日期、发件机构和主题以及原名称；模拟会加上 `(not moved)`，因为归档区中没有归入任何内容
`stays in place (not administrative)`	分拣把该文件放在一边；它既不移动也不重命名，每次处理都会再次列出
`(duplicate: would go to 重复/)`	文件指纹已经归档过；它进入 **重复/** 文件夹，而不是归档两次

领域 `incertain` 总是导向核查文件夹，模型意见不一致的文件也是如此。共有文件（身份证件、住址证明、银行账户信息）仅当所有归类模型返回同一类别时才建议放入各领域的保险库；否则进入核查文件夹。有疑问时，Dibab 宁可交给人来决定，也不冒险归档。

不用模型检查归档路径

要在不等待任何模型的情况下查看某个判定通向哪个目标，可给 `--dry-run` 提供一个把文件名对应到判定的 JSON 文件。程序对其应用归档路径表并显示目标位置。该选项只与 `--dry-run` 一起存在：与 `--execute` 同时使用时，程序以退出码 2 拒绝。

```
cat > verdicts.json <<'EOF'
{
  "notification-organisme.pdf": {"administratif": true, "domaine": "personnel", "categorie": "A
   ↪  ides", "personne": "Alice", "date": "2026-03-15", "emetteur": "Organisme public"},
  "recette-gateau.pdf": {"administratif": false, "categorie": "Inconnue"}
}
EOF
dibab --dry-run --verdicts-simules verdicts.json | sed -n '/Simulated routing/,$p'

model 1/1, document 1/3
model 1/1, document 2/3
model 1/1, document 3/3
Simulated routing (verdicts supplied, nothing is moved)
  - notification-organisme.pdf: domain personnel
      -> vault personnel, classement: ~/Classement/Personnel/Administratif/Alice/2026/2026-03-1
       ↪  5_Organisme-public_Aides.pdf
  - facture-hebergement.pdf: no simulated verdict
  - recette-gateau.pdf: not administrative, stays in place
  - courrier-sans-en-tete.pdf: no simulated verdict
```

归档：--execute

当模拟结果符合您的要求时，用 --execute 运行同一命令。每份文件复制到目标位置，副本用 SHA256 指纹校验，之后才从来源移除原件。在确认副本完全相同之前不移除任何内容。

```
dibab --execute

Dibab 0.9.0-alpha — session 2026-11-23T12:09:03
  Mode               : execute
  Configuration      : ~/.config/dibab/config.json
  Vault personnel     : ~/Classement/Personnel — mounted
  Vault professionnel : ~/Classement/Professionnel — mounted
  Session journal    : ~/.local/state/dibab/sessions/session_2026-11-23T12:09:03.jsonl
  Model ocr           : essai-ocr:latest
  Model tri_rapide    : essai-tri:latest
  Model classification: essai-classification:latest
  Consensus          : 1 classification model(s), threshold 0.4

Source: entrant (~/Classement/Entrant)
  - notification-organisme.pdf                          (928 B, mtime 2026-11-23)
  - facture-hebergement.pdf                             (922 B, mtime 2026-11-23)
  - recette-gateau.pdf                                  (895 B, mtime 2026-11-23)
  - courrier-sans-en-tete.pdf                           (901 B, mtime 2026-11-23)
  Source total: 4 file(s), 3.6 KB

============================================================
Scan total: 4 candidate file(s), 3.6 KB
============================================================
Text extraction (the text is neither shown nor written)
  - notification-organisme.pdf: text layer, 258 character(s), 1 page(s), 0.02 s
  - facture-hebergement.pdf: text layer, 246 character(s), 1 page(s), 0.02 s
  - recette-gateau.pdf: text layer, 226 character(s), 1 page(s), 0.02 s
  - courrier-sans-en-tete.pdf: text layer, 236 character(s), 1 page(s), 0.02 s

Filing (execution: each document is copied, checked by SHA256, then removed from its source)
model 1/1, document 1/3
model 1/1, document 2/3
model 1/1, document 3/3
  - notification-organisme.pdf: classe
      -> ~/Classement/Personnel/Administratif/Alice/2026/2026-03-15_Organisme-public_Aides.pdf
  - facture-hebergement.pdf: classe
      -> ~/Classement/Professionnel/Classement/2026/Factures/2026-03-15_Hebergeur-Exemple_Factu
      ↪  res.pdf
  - recette-gateau.pdf: not administrative, stays in place
  - courrier-sans-en-tete.pdf: a_verifier
```

```
      -> ~/Classement/Personnel/待核实/[待核实]courrier-de-suivi-de-dossier_courrier-sans-en-tete
      ↪  .pdf

Session report: ~/.local/state/dibab/sessions/rapport_2026-11-23T12:09:03.json
```

实际执行时，每行以所执行的操作结尾：`classe`（归入归档区）、`a_verifier`（放入核查文件夹）或 `doublon`（放入 **重复/** 文件夹）。这些词在所有语言中都相同，与日志中写入的操作一致。

文件现在位于报告所说的位置：

```
find ~/Classement -type f | sort

~/Classement/Entrant/recette-gateau.pdf
~/Classement/Personnel/Administratif/Alice/2026/2026-03-15_Organisme-public_Aides.pdf
~/Classement/Personnel/待核实/[待核实]courrier-de-suivi-de-dossier_courrier-sans-en-tete.pdf
~/Classement/Professionnel/Classement/2026/Factures/2026-03-15_Hebergeur-Exemple_Factures.pdf
```

三个文件离开了 `Entrant`。食谱留下了：它不是行政文件，Dibab 不触碰没有要求它归档的内容。

`--execute` 写入什么

每次移动都写入操作日志 `suivi.journal`，每个操作一行 JSON：

```
head -n 1 ~/.local/state/dibab/journal.jsonl

{"schema_version": 1, "session": "2026-11-23T12:09:03", "timestamp": "2026-11-23T12:09:03",
↪  "evenement": "action", "id_action": "<identifiant>", "action": "classe", "domaine": "person
↪  nel", "origine": "~/Classement/Entrant/notification-organisme.pdf", "empreinte_avant":
↪  "6850987e722babfbca9f474ce8971afc532762ac07cd767308c8294fc1383d28", "destinations":
↪  [{"chemin": "~/Classement/Personnel/Administratif/Alice/2026/2026-03-15_Organisme-public_Ai
↪  des.pdf", "empreinte_apres":
↪  "6850987e722babfbca9f474ce8971afc532762ac07cd767308c8294fc1383d28"}]}
```

该行包含操作及其标识符、领域、路径以及前后的指纹，不含文件文本。每次运行还在 `suivi.sessions` 文件夹中留下两个文件：运行日志（`session_<time>.jsonl`，与屏幕上相同的事件）和报告（`rapport_<time>.json`，包含各操作的计数、失败和耗时，不含文件名）。每个来源在 `dernier_scan.json` 中的扫描日期仅对无故障读取的来源向前推进。第 8 章说明如何控制这些文件的大小。

同一时间只运行一次实际执行。发现锁 `dibab.verrou` 已被占用的第二个 `--execute` 或 `--rollback` 以退出码 5 停止，不触碰任何内容。

撤销：--rollback

--rollback 撤销最近一次归档运行。--rollback --session <time> 撤销指定的运行，其时间即其开头打印并用于其文件名的时间。

```
dibab --rollback

Dibab 0.9.0-alpha — session 2026-11-23T12:09:03
  Mode               : rollback
  Configuration      : ~/.config/dibab/config.json
  Vault personnel     : ~/Classement/Personnel — mounted
  Vault professionnel : ~/Classement/Professionnel — mounted
  Session journal    : ~/.local/state/dibab/sessions/session_2026-11-23T12:09:03.jsonl

Rollback of session 2026-11-23T12:09:03: 3 undone, 0 failed
```

每份文件回到原来的位置，副本经指纹校验，目标文件被移除。日志中追加一行 annulation，日志从不改写。共有文件从各领域的保险库中一并移除。已经撤销过的操作不会执行两次。

```
find ~/Classement/Entrant -type f | sort

~/Classement/Entrant/courrier-sans-en-tete.pdf
~/Classement/Entrant/facture-hebergement.pdf
~/Classement/Entrant/notification-organisme.pdf
~/Classement/Entrant/recette-gateau.pdf
```

当某个操作无法撤销时，不为它移除任何内容，并显示原因，退出码为 1。当文件被手工编辑或移动时（放回原处或保持现状），原因是 destination modified since the filing 或 destination not found；当另一个文件占用了原位置时（移开它，然后重新运行命令），原因是 source occupied by another file。

退出码

退出码	含义
0	成功
1	找不到配置、配置无效或缺少键；至少一个来源或文件失败；撤销只部分完成；OAuth2 授权失败；--diagnose 至少有一项 ÉCHEC
2	在会写入的模式之前无法确认保险库已挂载；命令行无效
3	某个角色没有已安装的模型，或 Ollama 没有响应
4	本机拒绝执行 --execute（键 executer_sur）
5	另一次运行持有锁 dibab.verrou

--version 返回 0，命令行无效时返回 2。

保险库与加密

什么是保险库

保险库是接收某一领域文件的目录：个人领域一个，业务领域一个（`destinations.personnel.racine` 和 `destinations.professionnel.racine`）。Dibab 不创建、不挂载、不加密、不同步它。它期望该目录已经存在，并且由于文件敏感，期望它位于您在运行前打开的加密卷上。

危险很容易说清。加密卷关闭时，它的挂载点只是明文磁盘上的一个空目录。向那里写入的程序会以明文写下文件，处在加密层之下，且没有任何错误。因此 Dibab 在每次写入之前确认每个保险库确实已挂载。

挂载确认

在 `--execute`、`--rollback`、`--inventaire` 或 `--autoriser-google` 之前，Dibab 对每个保险库的根目录运行 `findmnt -n -T <root> -o FSTYPE`，把文件系统类型与该保险库的 `systemes_fichiers_acceptes` 列表比较。它还检查根目录可写。如果类型不在列表中，保险库未获确认，程序以退出码 2 停止，不触碰任何内容。

示例配置期望 `fuse.gocryptfs`，即 `findmnt` 对已挂载的 gocryptfs 卷报告的类型。如果其位置是一个普通目录（关闭的保险库看起来就是这样），会立即拒绝：

```
dibab --execute --config ~/.config/dibab/config-ferme.json; echo "exit code: $?"

ERROR: vault personnel: vault not mounted (type ext4, expected fuse.gocryptfs) (~/Classement/Pe
↪ rsonnel)
ERROR: vault professionnel: vault not mounted (type ext4, expected fuse.gocryptfs) (~/Classemen
↪ t/Professionnel)
       Nothing was touched. Mount the destination directory, then run again.
exit code: 2
```

模拟不会因保险库未获确认而停止：它在开头用 `[!]` 标出并继续，使您可以在打开保险库之前阅读建议。它不把运行日志写在保险库所包含的路径中，因为那会把日志以明文留在磁盘上。

选择接受的类型

保险库打开后，用第 2 章的命令读取您自己卷的类型，并写入列表：

```
"systemes_fichiers_acceptes": ["fuse.gocryptfs"]
```

如果您使用其他加密方式（LUKS、eCryptfs、其他工具的卷），请打开卷并写入 `findmnt` 为它返回的类型。在未加密的卷上，检查无法再区分未挂载与已挂载：您将因此失去这层保护，第 9 章的诊断会提醒您状态是明文。

gocryptfs 示例

gocryptfs 不是 Dibab 的一部分，Dibab 也不要求使用它。这是构建各领域保险库的一种方式。每个加密目录只创建一次，选择一个保存在机器之外的口令；命令会询问它。

```
mkdir -p ~/Chiffre/Personnel ~/Chiffre/Professionnel
gocryptfs -init ~/Chiffre/Personnel
gocryptfs -init ~/Chiffre/Professionnel
```

运行前打开每个保险库。命令再次询问口令，并把明文视图挂载到配置中指定的目录上：

```
gocryptfs ~/Chiffre/Personnel ~/Classement/Personnel
gocryptfs ~/Chiffre/Professionnel ~/Classement/Professionnel
```

运行后关闭它们：

```
fusermount3 -u ~/Classement/Personnel
fusermount3 -u ~/Classement/Professionnel
```

您备份和同步的是加密目录 `~/Chiffre/...`。Dibab 对它们一无所知。

状态存放在哪里

除文件之外，Dibab 还保存状态：每个来源上次扫描的日期、操作日志、运行日志和报告、已取回文件的登记簿以及锁。如果配置中没有任何 `suivi.*` 路径，这些都以明文放在 `~/.local/state/dibab/`（或 `$XDG_STATE_HOME/dibab/`）下，文件夹权限 `700`，文件权限 `600`。日志记录文件名和发件机构；这已经很能说明问题。诊断以警告标出：`AVERTISSEMENT emplacement_etat`。

要加密状态，请用 `suivi` 区块的三个键把它放进保险库：

```
"suivi": {
  "fichier_date": "~/Classement/Personnel/logs/dernier_scan.json",
  "journal": "~/Classement/Personnel/logs/journal.jsonl",
  "sessions": "~/Classement/Personnel/logs/sessions"
}
```

状态的其他文件（`rapatries.json`、`imap_deposes.json`、`dibab.verrou`）跟随 `fichier_date`。如此设置后，诊断不再报告警告。

磁盘空间

Dibab 只移动和复制，从不删除。归档的文件复制到目标位置、校验，然后从来源移除，因此归档不会使磁盘占用增长。与已归档文件重复的文件进入保险库的 **重复/** 文件夹，而不是归档两次；是否清空该文件夹由您决定，绝非工具决定。

通过 ssh 访问的来源

家庭中的远程机器通过 ssh 读取。Dibab 列出您指定目录中的文件，把它们复制到个人保险库的投放文件夹，用指纹校验每个副本，然后像本地文件一样归档。远程机器上的原文件保持原样：Dibab 从不在那里移除或修改任何内容。

远程机器必须提供什么

- 带 GNU find（列表使用 -printf）、sha256sum 和 cat 的 Linux 系统。
- 无需密码即可打开会话的 ssh 别名。有关连接的一切，即地址、用户和密钥，都在本地机器的 ~/.ssh/config 中该别名之下；Dibab 对此一概不知。例如，对一台虚构的机器：

```
Host machine1
    HostName 192.0.2.10
    User archive
    IdentityFile ~/.ssh/id_ed25519
```

首次运行前手动检查别名。该命令不应输出任何内容并返回 0：

```
ssh machine1 true
```

声明机器

涉及 config.json 的两个区块：给出时限的 ssh，以及列出机器的 sources_distantes。

```
"ssh": {
  "delai_connexion_secondes": 10,
  "delai_commande_secondes": 120,
  "delai_copie_secondes": 900
},

"sources_distantes": [
  {
    "id": "ssh-machine1",
    "alias_ssh": "machine1",
    "vide_poches_destination": "vide-poches-machine1",
    "repertoires": ["~/Téléchargements"],
    "date_reference": "modification",
    "recursif": true,
    "exclusions": []
  }
]
```

键	含义
ssh.delai_connexion_secondes	打开连接的时限（10）

键	含义
ssh.delai_commande_secondes	列表或指纹计算的时限（120）
ssh.delai_copie_secondes	复制一个文件的时限（900）
id	唯一名称；它是该机器在 dernier_scan.json 中的键
alias_ssh	~/.ssh/config 中的别名
vide_poches_destination	投放文件夹的名称，不含任何分隔符，创建在个人保险库根目录下；取回的文件到达这里
repertoires	远程机器上要读取的目录；~ 是远程用户的主目录
date_reference, recursif, exclusions	与本地来源相同（第 3 章）

一旦声明了远程来源，三个 ssh 时限即为必填。

首次运行：先列出，再取回

本章的配置只声明了该机器。模拟只列出文件，不复制任何内容：

```
dibab --dry-run --config ~/.config/dibab/config-ssh.json

Dibab 0.9.0-alpha — session 2026-11-23T12:09:03
  Mode               : dry-run
  Configuration      : ~/.config/dibab/config-ssh.json
  Vault personnel     : ~/Classement/Personnel — mounted
  Vault professionnel : ~/Classement/Professionnel — mounted
  Session journal    : ~/.local/state/dibab/sessions/session_2026-11-23T12:09:03.jsonl
  Model ocr           : essai-ocr:latest
  Model tri_rapide    : essai-tri:latest
  Model classification: essai-classification:latest
  Consensus          : 1 classification model(s), threshold 0.4

Remote source: ssh-machine1 (nothing is fetched in dry-run)
  - releve-bancaire.pdf                                (remote, 920 B, mtime 2026-11-23)

============================================================
Scan total: 0 candidate file(s), 0 B
============================================================
Text extraction (the text is neither shown nor written)

Classification (dry-run, nothing is moved; the text is neither shown nor written)
```

Remote source 区块用 remote 一词显示每个文件。投放文件夹中不写入任何内容。首次扫描时，只列出最近 suivi.premier_scan_jours 天（默认 90）内修改过、且扩展名在 extensions_cibles 中的文件。

实际执行会取回文件，然后归档：

```
dibab --execute --config ~/.config/dibab/config-ssh.json

Dibab 0.9.0-alpha — session 2026-11-23T12:09:03
  Mode               : execute
  Configuration      : ~/.config/dibab/config-ssh.json
  Vault personnel     : ~/Classement/Personnel — mounted
  Vault professionnel : ~/Classement/Professionnel — mounted
  Session journal    : ~/.local/state/dibab/sessions/session_2026-11-23T12:09:03.jsonl
  Model ocr           : essai-ocr:latest
  Model tri_rapide    : essai-tri:latest
  Model classification: essai-classification:latest
  Consensus          : 1 classification model(s), threshold 0.4

Transfer of remote sources (copy checked by fingerprint, nothing is removed on the remote side)
  ssh-machine1: 1 candidate(s)

Source: ssh-machine1-vide-poches (~/Classement/Personnel/vide-poches-machine1)
  - releve-bancaire.pdf                                (920 B, mtime 2026-11-23)
  Source total: 1 file(s), 920 B

============================================================
Scan total: 1 candidate file(s), 920 B
============================================================
Text extraction (the text is neither shown nor written)
  - releve-bancaire.pdf: text layer, 242 character(s), 1 page(s), 0.02 s

Filing (execution: each document is copied, checked by SHA256, then removed from its source)
model 1/1, document 1/1
  - releve-bancaire.pdf: classe
      -> ~/Classement/Personnel/Administratif/Bob/2026/2026-03-15_Banque-Exemple_Banques.pdf

Session report: ~/.local/state/dibab/sessions/rapport_2026-11-23T12:09:03.json
```

每个文件复制到 `vide-poches-machine1`，重新读回，并用 SHA256 与远程机器上计算的指纹比较。不匹配的副本不予保留。随后扫描把投放文件夹当作本地来源处理，不做任何日期过滤，因为投放文件夹随归档进行而清空：里面的内容就是在等待处理的内容。位于 `dernier_scan.json` 旁的登记簿 `rapatries.json` 按指纹记住已取回的文件。

再次运行

第二次执行不会取回任何新内容。远程文件再次被列出，但其指纹已在登记簿中，因此既不复制也不二次归档：

```
dibab --execute --config ~/.config/dibab/config-ssh.json

Dibab 0.9.0-alpha — session 2026-11-23T12:09:03
  Mode               : execute
  Configuration      : ~/.config/dibab/config-ssh.json
  Vault personnel     : ~/Classement/Personnel — mounted
  Vault professionnel : ~/Classement/Professionnel — mounted
  Session journal    : ~/.local/state/dibab/sessions/session_2026-11-23T12:09:03.jsonl
  Model ocr           : essai-ocr:latest
  Model tri_rapide    : essai-tri:latest
  Model classification: essai-classification:latest
  Consensus          : 1 classification model(s), threshold 0.4

Transfer of remote sources (copy checked by fingerprint, nothing is removed on the remote side)
  ssh-machine1: 1 candidate(s)

Source: ssh-machine1-vide-poches (~/Classement/Personnel/vide-poches-machine1)
  (no candidate file)
  Source total: 0 file(s), 0 B

============================================================
Scan total: 0 candidate file(s), 0 B
============================================================
Text extraction (the text is neither shown nor written)

Filing (execution: each document is copied, checked by SHA256, then removed from its source)

Session report: ~/.local/state/dibab/sessions/rapport_2026-11-23T12:09:03.json
```

该机器在 `dernier_scan.json` 中的日期只有在一次完整列表且所有副本均通过校验后才向前推进。失败的文件在下次执行时重新取回，不产生重复。同时存在于远程机器和本地扫描中的文件进入 **重复/** 文件夹：它只归档一次。

连接失败时

被拒绝时只停止相关机器；其他来源继续。Dibab 不反复重试，您也不应如此。

消息	可能原因
`Could not resolve hostname` 或 `Permission denied`	机器已关机，或别名或密钥错误；修正后再运行一次

消息	可能原因
超过时限	调高 `ssh.delai_commande_secondes`（列表）或 `ssh.delai_copie_secondes`（复制）
`remote directory partly readable or absent`	修正 `repertoires`
机器上有文件却没有候选	文件早于 `suivi.premier_scan_jours`，或扩展名不在 `extensions_cibles` 中

IMAP 邮箱与 Google 账户

Dibab 通过 IMAP 读取邮箱的附件，使用密码，或对 Google 账户使用 OAuth2 令牌。它只读不改：邮箱以只读方式打开，不把任何邮件标为已读，不设置任何标志，服务器上不复制、不移动、不删除任何内容。

向服务器请求什么

程序只向服务器发送 `EXAMINE`（以只读方式打开文件夹）、`UID SEARCH ... SINCE`（某日期以来收到的邮件），以及用于获取接收日期、邮件结构和每次一个附件的 `UID FETCH`，并使用 `BODY.PEEK`，即不把邮件标为已读的形式。从不请求邮件正文。连接始终使用 TLS，证书按系统的信任库校验：带 `"ssl": false` 的来源会被拒绝。

附件是邮件中带有名称且处置方式为 `attachment`，或为不带 `Content-ID` 的 `inline` 的部分（签名中的图像带有它，因而留在邮件中）。其扩展名必须在 `extensions_cibles` 中。其名称是发件人提供的数据：它会像取回文件的名称一样被解码和清理（无分隔符、无控制字符、绝不隐藏、长度受限），且不进入运行日志和报告。

声明邮箱

```
"imap": {
  "delai_secondes": 60,
  "taille_max_piece_mo": 25
},

"sources_imap": [
  {
    "id": "imap-foyer",
    "nom": "foyer",
    "serveur": "imap.example.com",
    "port": 993,
    "ssl": true,
    "authentification": "login",
    "utilisateur": "utilisateur@example.com",
    "mot_de_passe_env": "DIBAB_IMAP_FOYER",
    "dossiers": ["INBOX"],
    "vide_poches_destination": "vide-poches-foyer",
    "date_reference": "reception"
  }
]
```

键	含义
`imap.delai_secondes`	每次网络交互的时限，含连接（`60`）
`imap.taille_max_piece_mo`	按服务器公布的大小超过该值的附件不予取回（`25`）
`id`	唯一名称；它是该来源在 `dernier_scan.json` 中的键
`nom`	标签

键	含义
serveur, port, utilisateur	账户
ssl	必须为 true
authentification	login（密码放在环境变量中）或 oauth2（Google 账户）；其他任何值都会使该来源停止
mot_de_passe_env	保存密码的环境变量名（仅 login）
dossiers	要读取的文件夹，各以只读方式打开
vide_poches_destination	投放文件夹，单一名称，位于个人保险库根目录下
date_reference	reception：服务器记录的接收日期，绝不使用由发件人书写的 Date 头

两个区块 imap 和 sources_imap 必须同时存在；缺少某个键时，程序以退出码 1 停止并指出它。

密码

密码从不放在 config.json 中。Dibab 从 mot_de_passe_env 指定的环境变量读取它，从不写入、记录或显示它。请在启动 Dibab 的终端中定义它，不要写在任何地方：

```
read -rs DIBAB_IMAP_FOYER && export DIBAB_IMAP_FOYER
```

变量未定义的来源会带警告跳过，不建立任何连接。只要您保留 config.exemple.json 中示例条目的声明，它也会如此。

首次运行：先列出，再存放

本章的配置只声明了该邮箱。模拟列出将要取回的附件，带 imap 一词，且不下载任何附件：

```
dibab --dry-run --config ~/.config/dibab/config-imap.json

Dibab 0.9.0-alpha — session 2026-11-23T12:09:03
  Mode               : dry-run
  Configuration      : ~/.config/dibab/config-imap.json
  Vault personnel     : ~/Classement/Personnel — mounted
  Vault professionnel : ~/Classement/Professionnel — mounted
  Session journal    : ~/.local/state/dibab/sessions/session_2026-11-23T12:09:03.jsonl
  Model ocr           : essai-ocr:latest
  Model tri_rapide    : essai-tri:latest
  Model classification: essai-classification:latest
  Consensus          : 1 classification model(s), threshold 0.4

IMAP source: imap-foyer (nothing is downloaded in dry-run)
  - attestation-assurance.pdf                          (imap, 1274 B, received 2026-11-23)
```

```
============================================================
Scan total: 0 candidate file(s), 0 B
============================================================
Text extraction (the text is neither shown nor written)

Classification (dry-run, nothing is moved; the text is neither shown nor written)
```

实际执行下载每个附件，校验其指纹，放入投放文件夹，并像本地文件一样归档：

```
dibab --execute --config ~/.config/dibab/config-imap.json

Dibab 0.9.0-alpha — session 2026-11-23T12:09:03
  Mode               : execute
  Configuration      : ~/.config/dibab/config-imap.json
  Vault personnel     : ~/Classement/Personnel — mounted
  Vault professionnel : ~/Classement/Professionnel — mounted
  Session journal    : ~/.local/state/dibab/sessions/session_2026-11-23T12:09:03.jsonl
  Model ocr           : essai-ocr:latest
  Model tri_rapide    : essai-tri:latest
  Model classification: essai-classification:latest
  Consensus          : 1 classification model(s), threshold 0.4

IMAP sources (read-only: nothing is modified or deleted in the mailbox)
IMAP source: imap-foyer
  - attestation-assurance.pdf                          (imap, 1274 B, received 2026-11-23)

Source: imap-foyer-vide-poches (~/Classement/Personnel/vide-poches-foyer)
  - attestation-assurance.pdf                          (928 B, mtime 2026-11-23)
  Source total: 1 file(s), 928 B

============================================================
Scan total: 1 candidate file(s), 928 B
============================================================
Text extraction (the text is neither shown nor written)
  - attestation-assurance.pdf: text layer, 258 character(s), 1 page(s), 0.02 s

Filing (execution: each document is copied, checked by SHA256, then removed from its source)
model 1/1, document 1/1
  - attestation-assurance.pdf: classe
      -> ~/Classement/Personnel/Administratif/Alice/2026/2026-03-15_Assurance-Exemple_Assurance
       ↪  s.pdf

Session report: ~/.local/state/dibab/sessions/rapport_2026-11-23T12:09:03.json
```

位于 dernier_scan.json 旁的文件 imap_deposes.json 记住已存放的内容（文件夹、UIDVALIDITY、UID、部分和指纹）。指纹已在其中或已在操作日志中的附件不会再次存放：

```
dibab --execute --config ~/.config/dibab/config-imap.json

Dibab 0.9.0-alpha — session 2026-11-23T12:09:03
  Mode               : execute
  Configuration      : ~/.config/dibab/config-imap.json
  Vault personnel     : ~/Classement/Personnel — mounted
  Vault professionnel : ~/Classement/Professionnel — mounted
  Session journal    : ~/.local/state/dibab/sessions/session_2026-11-23T12:09:03.jsonl
  Model ocr           : essai-ocr:latest
  Model tri_rapide    : essai-tri:latest
  Model classification: essai-classification:latest
  Consensus          : 1 classification model(s), threshold 0.4

IMAP sources (read-only: nothing is modified or deleted in the mailbox)
IMAP source: imap-foyer
  - attestation-assurance.pdf                          (imap, 1274 B, received 2026-11-23)

Source: imap-foyer-vide-poches (~/Classement/Personnel/vide-poches-foyer)
  (no candidate file)
  Source total: 0 file(s), 0 B

============================================================
Scan total: 0 candidate file(s), 0 B
============================================================
Text extraction (the text is neither shown nor written)

Filing (execution: each document is copied, checked by SHA256, then removed from its source)

Session report: ~/.local/state/dibab/sessions/rapport_2026-11-23T12:09:03.json
```

该来源在 dernier_scan.json 中的日期只有在完整读取且每次存放都成功后才向前推进。

连接失败时

身份验证被拒绝或连接失败会停止该来源，不做第二次尝试。其他来源继续。

消息	可能原因
`environment variable ... not defined, source skipped`	在启动 Dibab 的终端中导出密码
`authentication refused by the server`	密码或用户错误；修正后再运行一次，不要循环重试

消息	可能原因
`connection impossible`	服务器、端口或网络问题
`folder ... not found`	修正 `dossiers`
`attachment(s) above imap.taille_max_piece_mo`	调高该值

使用 OAuth2 的 Google 账户

Google 账户使用 OAuth2 令牌而非密码读取。Google 控制台中的步骤需由您本人用要读取的 Google 账户完成，控制台的标签会随时间变化：概念以其 OAuth2 文档为准。

1. 在 Google Cloud 控制台 中创建项目，启用 Gmail API，配置 OAuth 同意屏幕（类型为“external”，联系人填您的地址）。然后发布应用，使其状态为“In production”。在“Testing”状态下，授权七天后过期。
2. 创建类型为“Desktop application”的 OAuth 凭据并下载其 JSON 文件。
3. 把该文件保存在保险库中权限为 `700` 的文件夹里，文件权限 `600`。Dibab 从不修改它。它就是 `client_fichier`。
4. 用 `"authentification": "oauth2"`、`client_fichier` 和 `token_fichier` 代替 `mot_de_passe_env` 声明来源，并添加 `oauth2` 区块：

```
"oauth2": {
  "url_autorisation": "https://accounts.google.com/o/oauth2/v2/auth",
  "url_jeton": "https://oauth2.googleapis.com/token",
  "portee": "https://mail.google.com/",
  "marge_renouvellement_secondes": 300,
  "delai_secondes": 30,
  "delai_consentement_secondes": 300,
  "chemins_interdits": ["~/Documents"]
},

"sources_imap": [
  {
    "id": "imap-google",
    "nom": "foyer-google",
    "serveur": "imap.gmail.com",
    "port": 993,
    "ssl": true,
    "authentification": "oauth2",
    "utilisateur": "utilisateur@example.com",
    "client_fichier": "~/Classement/Personnel/logs/client-google.json",
    "token_fichier": "~/Classement/Personnel/logs/jeton-google.json",
```

```
    "dossiers": ["INBOX"],
    "vide_poches_destination": "vide-poches-google",
    "date_reference": "reception"
  }
]
```

oauth2 的键	含义
`url_autorisation`, `url_jeton`	Google 的接口；除本地回环外必须为 `https`
`portee`	OAuth 权限范围（`https://mail.google.com/`）
`marge_renouvellement_secondes`	访问令牌剩余时间少于该值时续期（`300`）
`delai_secondes`	与授权服务器每次交互的时限（`30`）
`delai_consentement_secondes`	`--autoriser-google` 等待您应答的时长（`300`）
`chemins_interdits`	禁止存放令牌的文件夹（`~/Documents`）

5. 在机器上的终端中用该来源的 `id` 授权一次：

```
dibab --autoriser-google imap-google
```

Dibab 输出以 `https://accounts.google.com/` 开头的地址。在浏览器中打开它。接受“unverified application”警告（因为应用是您自己的，所以会显示），然后同意授权。程序在本地回环上接收应答，进行检查，换取令牌，并在权限为 `700` 的文件夹中以 `600` 权限写入令牌文件。令牌文件必须位于任何 git 仓库和 `chemins_interdits` 之外；可被组或其他用户读取的令牌文件会使来源停止。如果拒绝授权、未及时收到应答，或 Google 没有返回续期令牌，则不写入任何内容；请重新运行该命令。

此后扫描像密码来源一样运行，无需任何环境变量：访问令牌自动续期，令牌既不出现在日志中也不显示在屏幕上。令牌文件不存在不是错误：该来源会被跳过，并给出需要运行的命令。如果 Google 撤销授权（您移除了应用的访问权限、修改了 Google 账户密码，或六个月没有扫描），Dibab 以明确消息停止该来源，您需要重新运行 `--autoriser-google`。

权限范围 `https://mail.google.com/` 是 Google 为 IMAP 记载的唯一范围，它允许读取、写入和删除。邮箱保持不被修改，是由 Dibab 发送的内容保证的，即只发送本章开头列出的命令，而非由权限范围保证。第 10 章会再次说明。

清单与轮转

读取保险库中已有的内容：--inventaire

如果保险库中已经有手工或由其他工具归档的文件，`--inventaire` 会读取它们并写出迁移计划，不移动、不重命名、不删除任何内容。它是只读的：之前和之后，每个保险库完全相同。它需要第 5 章的挂载确认，因为每份计划写在其保险库中。

为了有内容可读，请再次归档试验区的四份文件；第 4 章的撤销已把它们放回来源：

```
dibab --execute | tail -n 9
```

```
model 1/1, document 1/3
model 1/1, document 2/3
model 1/1, document 3/3
  - notification-organisme.pdf: classe
      -> ~/Classement/Personnel/Administratif/Alice/2026/2026-03-15_Organisme-public_Aides.pdf
  - facture-hebergement.pdf: classe
      -> ~/Classement/Professionnel/Classement/2026/Factures/2026-03-15_Hebergeur-Exemple_Factu
      ↪  res.pdf
  - recette-gateau.pdf: not administrative, stays in place
  - courrier-sans-en-tete.pdf: a_verifier
      -> ~/Classement/Personnel/待核实/[待核实]courrier-de-suivi-de-dossier_courrier-sans-en-tete
      ↪  .pdf

Session report: ~/.local/state/dibab/sessions/rapport_2026-11-23T12:09:03.json
```

然后读取各领域的保险库：

```
dibab --inventaire
```

```
Dibab 0.9.0-alpha — session 2026-11-23T12:09:03
  Mode               : inventaire
  Configuration      : ~/.config/dibab/config.json
  Vault personnel     : ~/Classement/Personnel — mounted
  Vault professionnel : ~/Classement/Professionnel — mounted
  Session journal    : ~/.local/state/dibab/sessions/session_2026-11-23T12:09:03.jsonl

Inventory of what exists (read-only, nothing is moved)
  Vault personnel     : 3 listed document(s), 3 in the filed zone, 0 to file, 0 duplicate(s), 0
   ↪  present in the other vault, 1 in 待核实 (not listed)
      plan written: ~/Classement/Personnel/logs/inventaire_2026-11-23T12:09:03.json
  Vault professionnel : 1 listed document(s), 1 in the filed zone, 0 to file, 0 duplicate(s), 0
   ↪  present in the other vault, 0 in 待核实 (not listed)
      plan written: ~/Classement/Professionnel/logs/inventaire_2026-11-23T12:09:03.json
```

对每个保险库，报告给出读取的文件数、其中已在归档区的数量、位于归档区之外且应归档的数量、与同一保险库中另一文件重复的数量，以及同时存在于另一保险库的数量。计划是一个 JSON 文件 `logs/inventaire_<time>.json`，写在它所描述的保险库中：

```
ls ~/Classement/Personnel/logs

inventaire_2026-11-23T12:09:03.json
```

每个保险库的计划只列出该保险库的文件。关于另一个保险库，它只保留一个事实“该指纹在那里也存在”，因此一个保险库的计划从不包含另一个保险库中文件的路径。

计划中的每份文件都会得到一条建议：

建议	含义
无需处理	文件在归档区
放入投放文件夹，然后阅读模拟结果	文件位于归档区之外：把它放进某个来源，由下一次处理进行归类
与另一文件相同：核实后归入 `重复/`	指纹已经存在

`inventaire.exclusions` 中列出的文件夹（示例中为 `logs`、`Archive`、`rebut` 以及 `destination.repertoire_doublon` 指定的文件夹）、核查文件夹、投放文件夹和隐藏文件夹不会读取。

控制日志大小：轮转

有两处文件随使用而增长：操作日志 `suivi.journal` 和运行文件夹 `suivi.sessions`。当 `--execute` 或 `--rollback` 启动时，轮转应用 `suivi.rotation` 的限制：

键	作用
`journal_taille_max_mo`	日志超过该大小时重命名为 `journal.1.jsonl`，较旧的归档编号各加一，并开始新日志
`journal_archives_max`	保留的归档数量；新归档超出时移除最旧的
`sessions_max`	保留的运行日志和报告数量；移除最旧的

这是 Dibab 执行的唯一删除操作，它只涉及自己的日志，从不涉及文件。仍含有未撤销操作的归档从不移除：轮转宁可等待，也不丢失 `--rollback` 必须能够读取的内容。对某次归档有疑虑时，请在其日志变旧之前撤销。

状态文件以 `600` 权限写入，其文件夹权限为 `700`。它们保存在哪里，是第 5 章所述的选择。

诊断

`--diagnose` 检查安装、配置、Ollama 和各领域的保险库，每项检查一行，不写入任何内容：不创建文件、不移动任何东西、不占用锁。它只建立一个连接，即到 Ollama，用于读取模型列表。它不连接任何远程机器或邮箱，因此随时运行都是安全的。无论前面检查结果如何，它都会把每项检查执行到底。

```
dibab --diagnose

OK python : 3.13.5
OK outil_pdftotext : /usr/bin/pdftotext
OK outil_pdftoppm : /usr/bin/pdftoppm
OK outil_findmnt : /usr/bin/findmnt
OK outil_ssh : /usr/bin/ssh
OK configuration : ~/.config/dibab/config.json
AVERTISSEMENT emplacement_etat : the state and the journal are in clear text, outside any vault:
 ↪  ~/.local/state/dibab
OK ollama : http://127.0.0.1:18434 answers, 3 model(s) installed
OK modele_ocr : essai-ocr:latest
OK modele_tri_rapide : essai-tri:latest
OK modele_classification : essai-classification:latest, 1 model(s) in the consensus
NON_CONFIGURÉ modele_tri_decision : ollama.roles.tri_decision absent or empty: the triage goes
 ↪  through ollama.roles.tri_rapide
OK destination_personnel : ~/Classement/Personnel
OK destination_professionnel : ~/Classement/Professionnel
Diagnostic: 12 OK, 1 warning(s), 0 failure(s), 1 not configured, 0 not checked
```

每行格式为 `<STATE> <identifier> : <message>`，最后以汇总结束报告。消息使用您会话的语言；状态和标识符从不随语言变化，便于脚本比较。

状态

状态	含义
`OK`	检查通过
`AVERTISSEMENT`	不妨碍任何模式的缺陷，或本实例不使用的组件的缺陷
`ÉCHEC`	安装或所找到的配置所要求的某项检查失败
`NON_CONFIGURÉ`	该检查读取的配置尚不存在
`NON_VÉRIFIÉ`	该检查依赖另一项未通过的检查，或依赖缺失的配置

没有 `ÉCHEC` 时退出码为 `0`，至少有一个时为 `1`。警告、缺少配置和未检查项不改变它。

按顺序列出的检查

标识符	检查内容	若不是 OK
python	解释器为 3.10 或更高版本	低于时为 ÉCHEC
outil_pdftotext, outil_pdftoppm, outil_findmnt	工具在 PATH 中可执行	缺失时为 ÉCHEC
outil_ssh	ssh 在 PATH 中可执行	配置声明了 sources_distantes 时为 ÉCHEC，否则为 AVERTISSEMENT
configuration	找到配置（第 2 章）且有效：JSON 可读，具备每种模式要求的键	未找到：NON_CONFIGURÉ，消息列出尝试过的四个位置；找到但无效：ÉCHEC，并给出缺失的键
emplacement_etat	状态和日志位于某个已声明保险库的根目录下	位于任何保险库之外：AVERTISSEMENT（状态为明文）
ollama	GET <ollama.url>/api/tags 在 ollama.timeout_secondes 内应答	无应答：ÉCHEC
modele_ocr, modele_tri_rapide, modele_classification	该角色的首选项中至少有一个已安装的模型	没有安装或列表为空：ÉCHEC，消息指出 ollama.roles.<role>
modele_tri_decision	可选的决策分拣（第 3 章）：ollama.roles.tri_decision 的第一个已安装模型、decision 能力、classification.seuils_tri_decision 中的有效阈值、ollama.roles.objet 的已安装模型	角色缺失或为空：NON_CONFIGURÉ；已声明但无法使用：AVERTISSEMENT，消息指出需要修改的键，分拣改用 tri_rapide
destination_personnel, destination_professionnel	第 5 章的挂载确认：根目录的文件系统类型被接受且根目录可写	未确认：ÉCHEC，消息给出原因

带密码的来源若其环境变量未定义，会在 configuration 检查中以警告报告，因为程序运行时会跳过这样的来源。任何检查都不写入机密、文件名或环境变量的内容。

阅读失败结果

试验区的某个配置为每个领域声明了必须是 `fuse.gocryptfs` 的保险库，而这些目录是普通目录，这正是关闭的保险库的样子：

```
dibab --diagnose --config ~/.config/dibab/config-ferme.json; echo "exit code: $?"

OK python : 3.13.5
OK outil_pdftotext : /usr/bin/pdftotext
OK outil_pdftoppm : /usr/bin/pdftoppm
OK outil_findmnt : /usr/bin/findmnt
OK outil_ssh : /usr/bin/ssh
OK configuration : ~/.config/dibab/config-ferme.json
AVERTISSEMENT emplacement_etat : the state and the journal are in clear text, outside any vault:
↪  ~/.local/state/dibab
OK ollama : http://127.0.0.1:18434 answers, 3 model(s) installed
OK modele_ocr : essai-ocr:latest
OK modele_tri_rapide : essai-tri:latest
OK modele_classification : essai-classification:latest, 1 model(s) in the consensus
NON_CONFIGURÉ modele_tri_decision : ollama.roles.tri_decision absent or empty: the triage goes
↪  through ollama.roles.tri_rapide
ÉCHEC destination_personnel : vault not mounted (type ext4, expected fuse.gocryptfs) (~/Classem
↪  ent/Personnel)
ÉCHEC destination_professionnel : vault not mounted (type ext4, expected fuse.gocryptfs) (~/Cla
↪  ssement/Professionnel)
Diagnostic: 10 OK, 1 warning(s), 2 failure(s), 1 not configured, 0 not checked
exit code: 1
```

两项 `destination_*` 检查失败，退出码为 `1`，而 Ollama、模型和工具正常。消息说明应做什么：打开保险库，或在 `systemes_fichiers_acceptes` 中写入正确的类型。

其他常见失败：

行	应做什么
`ÉCHEC ollama`	服务无应答：检查 `ollama.url`，然后确认 Ollama 正在运行（`systemctl status ollama`）
`ÉCHEC modele_<role>`	按 `ollama list` 修改 `ollama.roles.<role>`
`AVERTISSEMENT modele_tri_decision`	安装所指的决策模型或主题模型，或写入其阈值；在此之前分拣改用 `tri_rapide`
`ÉCHEC destination_<domain>`	打开所指的保险库，或修改接受的类型
`ÉCHEC configuration`	从 `config.exemple.json` 复制区块，补上所指的键

全新安装

安装刚完成、既没有配置也没有 Ollama 时，诊断不算失败：缺少配置是全新安装的正常状态，标为 NON_CONFIGURÉ，不与故障混淆。在空的主目录中：

```
HOME="$(mktemp -d)" dibab --diagnose; echo "exit code: $?"

OK python : 3.13.5
OK outil_pdftotext : /usr/bin/pdftotext
OK outil_pdftoppm : /usr/bin/pdftoppm
OK outil_findmnt : /usr/bin/findmnt
OK outil_ssh : /usr/bin/ssh
NON_CONFIGURÉ configuration : no configuration found (the --config option: not given; the DIBAB
↪  _CONFIG variable: not given; the XDG location: ~/.config/dibab/config.json; the script
↪  folder: ~/applications/dibab/config.json)
NON_VÉRIFIÉ emplacement_etat : depends on the configuration
NON_VÉRIFIÉ ollama : depends on the configuration
NON_VÉRIFIÉ modele_ocr : depends on the configuration
NON_VÉRIFIÉ modele_tri_rapide : depends on the configuration
NON_VÉRIFIÉ modele_classification : depends on the configuration
NON_VÉRIFIÉ modele_tri_decision : depends on the configuration
NON_VÉRIFIÉ destination_personnel : depends on the configuration
NON_VÉRIFIÉ destination_professionnel : depends on the configuration
Diagnostic: 5 OK, 0 warning(s), 0 failure(s), 1 not configured, 8 not checked
exit code: 0
```

工具为 OK，配置为 NON_CONFIGURÉ，依赖它的八项检查为 NON_VÉRIFIÉ。退出码为 0。一旦存在配置，凡是它声明而机器不具备的都是 ÉCHEC：如果在 Ollama 停止时诊断仍返回 0，就会掩盖您正要查找的故障。

使用另一种语言

语言依次取自 LC_ALL、LC_MESSAGES、LANG：

```
LC_ALL=fr_FR.UTF-8 dibab --diagnose | tail -n 1

Diagnostic : 12 OK, 1 avertissement(s), 0 échec(s), 1 non configuré(s), 0 non vérifié(s)
```

自主性与局限

工具保证什么

- **任何文件都不离开您的网络**。对语言模型的调用是发往 `ollama.url`（默认 `http://localhost:11434`）的普通 HTTP 请求。文件和文件名不会发往任何其他地方，也没有遥测。
- **代码能访问的外部服务只有您声明的那些：**通过 `ssh` 的机器、IMAP 邮箱，以及启用 OAuth2 时 Google 的授权接口。
- **`config.json` 中不保存任何机密**。邮件密码放在环境变量中，Google 令牌放在任何仓库之外权限为 `600` 的文件中。令牌从不写入日志、显示在屏幕上或放入错误消息。
- **归档绝不会丢失文件**。文件先复制，副本用 SHA256 校验，之后才移除原件。每次归档都写入操作日志，并可用 `--rollback` 撤销。
- **邮箱和远程机器保持原样**。发送的 IMAP 命令就是第 7 章列出的那些；远程机器只被读取，那里不移除任何内容。
- **文件文本是数据，绝不是指令**。它由随机标记界定，模型以封闭的 JSON 作答并由 Dibab 校验。不符合模式的应答计为“不确定”，导向核查文件夹。模型永远无法触发任何操作。
- **有疑问时交给人来决定**。领域不确定、模型意见不一，或共有文件没有一致的类别时，文件放入 `待核实/`，而不是冒险的位置。
- **日志记录路径和类别，从不记录文本**。但它确实记录文件名和发件机构，这些已经很能说明问题；请把它保存在保险库中（第 5 章）。

代码以 GNU Affero 通用公共许可证第 3 版发布。它只使用 Python 标准库，因此没有需要审计的第三方软件包。

尊重家庭成员

只有在当事人明示或默示同意的情况下，一台机器、一个目录或一个邮件账户才纳入工具范围。软件无法对此进行检查：实例的管理员对此作出承诺。不要让 Dibab 指向不知道自己文件或邮箱被读取的人。

局限

- **远程来源只针对虚假服务器试验过**。 `ssh` 来源以及使用密码或 OAuth2 的 IMAP 来源由测试和本手册的重放覆盖，它们使用虚假的 `ssh`、虚假的 IMAP 服务器和虚假的授权服务器。在家庭中真实机器和真实邮箱上的使用（包括与 Google 的授权同意）尚待进行：在完成该试验之前，请把这些来源视为尚未在实际条件下经过检验，并从 `--dry-run` 开始。
- **归档质量取决于您的模型**。在参考实例上，对 15 份真实文件的首次试验归档了 2 份，将 3 份判为非行政文件搁置，另有 10 份出于谨慎放入了 `待核实/`。信任之前请先测量；源代码中的测量工具 `_outils/mesurer_domaines.py` 对已归档文件抽样，并把模型判定与各文件所在的保险库进行比较。
- **仅支持 Linux**。挂载确认依赖属于 `util-linux` 的 `findmnt`。
- **Dibab 不挂载、不加密、不同步任何内容，**也不管理磁盘空间或备份：这些由您负责。
- **Google 的权限范围大于实际用途**。权限范围 `https://mail.google.com/` 是 Google 为 IMAP 记载的唯一范围。它允许读取、写入和删除。邮箱保持不被修改，依靠的是 Dibab 发送的命令（列在第 7 章），而非权限范围。

- **默认值只是示例**。 `config.exemple.json` 中的类别、已知发件机构和命名模板是通用的；请按您所在国家调整。
- **Google 控制台中的步骤取决于 Google 的界面，**界面会变化。概念仍以其 OAuth2 文档为准。
- **项目尚年轻**。版本为 0.9.0-alpha；没有持续集成。源代码测试需手动运行。
- **Ollama 不是 Dibab 的一部分**。 Dibab 从不安装或更新它；上游改名或移除的模型，需要在 `ollama.roles` 中修改相应行。

报告问题

漏洞请按源代码中的 `SECURITY.md` 文件私下报告，切勿在公开议题中提交。参与贡献请阅读同一位置的 `CONTRIBUTING.md`。