DeepSeek Harness 包

- 包工作区
- core/ 产品 API 主干: 会话、提示词、工具、agent 服务与具体循环
- api/ Remote BFF 装配与 Typert RPC 网关
- typert/ 类型图生成、产物加载与运行时注册表
- goal/ 同会话 goal 的持久化与生命周期
- schedule/ — 仅限会话内的提醒
- feedback/: 记录的人类反馈
- identity/ — 共享身份
- llm/ — LLM 能力家族
- e2b/ E2B 远程运行时提供方
- subprocess/: 子进程能力家族
- shell/ — bash 能力家族
- terminal/: 持久 PTY 能力家族
- code-runtime/——代码执行能力族
- sandbox/ 进程限制 seam; bwrap/Landlock/Seatbelt 后端
- fs/ 文件系统能力系列: seam、本地实现、面向模型的文件工具、发现工具
- lsp/: 语言服务器代码导航
- skill/ — skill（技能）能力家族
- compaction/ — 压缩能力家族
- context/ — 请求上下文插件
- subagent/ subagent 能力系列: 提供方注册表约定和面向模型的委托工具
- jobs/: 后台任务能力家族
- experimental/ 私有原型与内部专用插件
- workflow/ 工作流 seam，worker 线程引擎、面向模型的 workflow/ralph 工具
- webhook/ — 从已验证外部事件到 DSH Session
- web/: web 访问能力家族
- attachment/: 持久附件能力族

- spill/ spill
- todo/ todo_write
- plan/ plan
- preset/ preset cordis.yml agent
- guard/ guard
- bundle/ profile
- extensions/ agent /
- hooks/ + Claude Code / Codex
- session/ —
- session-query/
- settings/
- credentials/
- storage/ + +
- workspace/ Workspace
- sdk/ Harness
- acp/ — Agent Client Protocol
- interaction/
- boot/ app bin
- host/ — Web GUI
- client/ — Web GUI
- test-support/ testkit Loader
- runtime-diagnostics/
- util/

包工作区

---
description: "DeepSeek Harness 包工作区: packages/ 下的 npm 包如何分组、每个组负责什么，以及约束它们的约定。"
kind: "package-group"

包

English | 中文

概述

harness 的 `packages/` 下的 npm 包组装而成，按能力系列分组：会话与 agent 循环、面向模型的工具、s `@deepseek-ai/dsh-*` 两作与 subagent；等等组脚本

目录

- 包分组
- 发布预期
- 依赖
- 包 README 约定
- 开发备注

包分组

每个包只属于一个组；新包加入现有组，新组则更新其自身 README 与本表。

组	职责

core/	产品 API 主干: 会话、提示词、工具、agent 服务与具体循环
api/	Remote BFF 装配与 Typert RPC 网关
typert/	类型图生成、产物加载与运行时注册表
goal/	同会话 goal 的持久化与生命周期
schedule/	仅限会话内的定时后续操作
feedback/	人类反馈的采集与命令
identity/	共享匿名身份
llm/	LLM 能力系列: 抽象服务 + 提供方适配器
e2b/	E2B 远程运行时提供方
subprocess/	子进程能力系列: Service Definition + 本地进程树提供方
shell/	Bash 能力系列: 执行器 seam、本地实现、面向模型的工具
terminal/	持久 PTY 能力系列: 限定所有者范围的会话、本地实现、面向模型的工具
code-runtime/	代码执行能力系列: Service Definition + worker 线程提供方 + PTC mode Consu
sandbox/	进程限制 seam; bwrap/Landlock/Seatbelt 后端
fs/	文件系统能力系列: seam、本地实现、面向模型的文件工具、发现工具
lsp/	LSP 能力系列: seam、lsp d 提供方和
skill/	skill 能力系列: 提供方注册表、本地提供方、面向模型的目录/loader
compaction/	压缩能力系列: Service Definition + 基础提供方 + 命令 Consumer
context/	模型可见请求上下文: workspace 指令、时间上下文、引用
subagent/	subagent 能力系列: 提供方注册表约定和面向模型的委托工具
jobs/	通用后台任务运行时和面向模型的作业控制工具
experimental/	私有原型与内部专用插件
workflow/	工作流 seam、work workflow ralph 工具
webhook/	已验证外部事件、受信规则与即发即弃 Workspace Session
web/	Web 能力系列: seam、搜索/获取提供方、面向模型的 Web 工具
attachment/	持久附件标识、校验、本地内容寻址存储
spill/	spill 能力系列: 存储 seam、本地实现、工具结果 spill 策略
todo/	todo_write 工具
plan/	Plan 协作状态，提供直接进入命令与经评审的退出
preset/	由 preset cordis.yml 按会话组装 agent
guard/	循环 tools/execute 截止时间强制执行器
bundle/	dsh --profile 补丁层
extensions/	agent 运行时自修改: 实时插件/服务检查与模型所写挂载/卸载

hooks/	钩子桥接 + 共享的 Claude Code / Codex 线协议库
session/	持久会话数据平面：持久化 seam + 后端、投影 seam、基于日志的标题、会话上报
session-query/	会话检索系列：逻辑语料库、有界读取、血缘、语义过滤、SQLite 全文搜索
settings/	用户设置 seam + 基于文件的提供方
credentials/	凭据引用/记录 sea .env 项提供方；用于询问人类的授权 flow
storage/	非会话存储中枢 + 后端 + 领域形式
workspace/	Workspace 实体
sdk/	进程外 SDK：JSON-RPC 协议与 TypeScript 客户端/服务器
acp/	仅面向自动化的 Agent Client Protocol 服务器
interaction/	人机协作平面：批准/交互 seam、权限预设、命令、询问用户的工具
boot/	共享的 app bin 启动粘合层
host/	web GUI 宿主半侧：API 网关 + HTTP 路由服务器
client/	web GUI 浏览器半侧：sh ui-* 组件对象服务、slot
test-support/	支持基础设施（testkit、不变式、回放、Loader 冒烟测试）
runtime-diagnostics/	运行时诊断：按包归属的运行时不变式检查与报告
util/	Branded<B> home/路径辅助函数、超时、留存）

发布预期

大多数组是产品– e2b/ API experimental/ test-support/ runtime-diagnostics/ util/ 是兼容性预期较低的支持组。

依赖

docs/module-graph.md pnpm run gen-module-graph，CI 中有新鲜度门禁）。

扩展插件依赖 **Service Defin** dsh-agent-loop 而 dsh-agent 则链接可以依赖主干插件。能力在需要独立演进时分离 Service Definition / Service seam。

□ README □□

□□□ README □ `## Known Limitations and Deferred Work` □□□□□——□□□□

core/ 产品 API 主干：会话、提示词、工具、ager

---
description: "core 分组地图：构成产品 API 主干的会话日志、系统提示词组装、工具注册表、agent 词汇与默认循环。"
kind: "package-group"

English | 中文

概述

core 分组提供 DeepSeek Harness [Agent] 句柄：收集驱动会话和具体循环所需的每个组装部分的最小包，插 dsh-base 赖的正是它们稳定的约定。一个轮次会流经其中全部环节——

目录

- 包
- 相关文档
- 开发备注

包

包	职责	ctx key
scope/	隔离单个 agent 贡献的作用域注册与事件路由	库，不使用 ctx key
session/	每个 agent 的历史都由其派生的仅追加会话事件日志	ctx.sessions
system-prompt/	由有序段、工具 schema 与变量进行的系统提示词组装	ctx.systemPrompt
tools/	循环分发所经过的工具注册表与受守卫的执行流水线	ctx.tools
agent-tool-presentation/	为 preset 提供按 agent 的工具呈现方式选择器	无 ctx key
agent/	[Agent] 句柄，以及其实时注册表与事件	ctx.agents
agent-default-model/	入口对全新 agent 应用的部署默认模型选择	ctx.agentDefaultModel

包	职责	ctx key
agent-loop/	默认 agent 驱动器：创建 agent 并运行轮次与步骤生命周期	ctx.agentLoop

scope agent Agent [agent-loop [agent agent-default-model [packages/bundle 的聚合层，拥有运行组合于组件。

相关文档

- Core 子系统 [Agent 句柄约定。
- 会话子系统的会话事件词汇与派生历史。
- 系统提示词子系统、动态上下文与工具 schema 类型。
- 工具子系统的工具执行流水线与呈现词汇。
- 作用域注册与查找表所依赖的作用域层原语。
- 架构——轮次流与新行为归属。
- 基础组合的默认产品组合。
- SDK 最小组合的完整、独立且功能集更小的组合。

开发备注

维护者的工作上下文——点击展开

无。

api/ Remote BFF □□□ Typert R

□□

---
description: "□□ Remote □□□□□□□□□ Client □ Host □□□□□□□□□□□□□□□□□□□□□□□□□□□□□"
kind: "package-group"

English | □□

□□

api/ □□□□□□ Remote □□Client □□□□□□□□□ Hc remotes □□□□□□□□ gateway □□□□□□□□□□□□□□□□□□□□

包	职责	ctx key
settings-controller/	拥有 settings 域各 seam 之上的配置界面读写。	ctx.settingsController、ctx.credentialsController / ctx.remote.settings ctx.remote.credentials
workspace-controller/	拥有 Workspace 变更与完整 Client Workspace 投影。	ctx.workspaceController / ctx.remote.workspace

Remote 调用沿 Client → Host 方向运行在应用共享的 Connection 之上。API Gateway 拥有 Remote 传输，各 controller 包分别拥有 Sess
Connection Fetch 路由。

相关文档

先读 API Gateway 参考以端到端了解 Remote 模型，再读 Typert 子系统页了解共享定义，并通过 Connection 了解物理载体。

- API Gateway ⊞⊞—Typert API Gateway 的现状参考：编程模型、生成流水线与运行时调用。
- Typert ⊞⊞⊞protocol、Gateway 与消费方装配共享的公共约定。
- Connection——每次 Remote 调用背 /api P[任⊞⊞与响应封装。

开发备注

维护者的工作上下文——点击展开

无。

typert/ 类型图生成、产物加载与运行时注册表

---
description: "typert 组地图: 构建时类型图生成器、运行时注册表、Loader 集成与共享 Remote 协议，它们共同支撑类型化的 Host 到 Client 调用。"
kind: "package-group"

English | 中文

概述

借助 typert 组，Client 环境能以类型化方法调用 Host 能力，并在无需手写协议代码的情况下共享生成的 schema 与反射信息。构建时生成器把源代码类型声明转换为与编译器无关的模型与运行时注册表，Loader 集成与共享 Remote 协议，它们共同支撑类型化的 Host 到 Client 调用。本 README 负责各自的配置、用法与限制。

目录

- 包
- 相关文档
- 开发备注

包

包	职责	**ctx** 键
generator/	在构建时分析源代码类型，并生成运行时加载所需的反射、schema 与 Remote 描述符	
loader/	把 Loader 组合中的生成 Typert 产物自动注册到运行时注册表	[ctx.loader [ctx.typert
protocol/	声明 Host 与 Client 共享的 Remote 装饰器、wire 描述符、编译器与提供方约定	
registry/	在运行时保存生成的包反射与实时 Zod schema，并提供 lookup 与 (ctx.typert 方注册表	

相关文档

- Typert 子系统参考协议与注册表类型记录的字面公共约定。
- API Gateway 参考——生成的 Remote 描述符如何成为实际的 Host 到 Client 调用。
- Remote 调用 Agent Note——Remote 调用背后的架构与传输决策。
- 包工作区地图区中的每个组及其职责。

开发备注

维护者的工作上下文——点击展开

无。

goal/ □□ goal □□□□□□□□

---
description: "goal □□□□□□□□

相关文档

- goal 子系统–goal [goal/change 事件与生成的服务 API。
- 生成的工具型录收的三个 goal 工具 schema。
- 生成的配goal 服务的每个受支持配置字段。
- goal 领域 Agent Note——领域设计及其决策。
- 同会话驱动器 Agent Note竞态与生命周期理由。

开发备注

维护者的工作上下文——点击展开

无。

schedule/ — □□□□□□□

---

description: "schedule □□□□□□□

- 仅限会话内的 Schedule 长记系统转换、视图与交付约定。
- 生成的工 `schedule_create` `schedule_list` `schedule_delete` schema。
- Schedule 用户指南挂载本包的官方配置路径。
- Web Schedule 目录——活动记录的可选只读浏览器呈现。

开发备注

维护者的工作上下文——点击展开

无。

feedback/□□□□□□□

---

description: "feedback □□□□□□□ assistant □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"

kind: "package-group"

English | □□

□□

feedback □□□□□□ harness □□□□□□□□ `/feedback` □ `messageFeedback` assistant □□□□□□□□□□□□□□□□□□□□

相关文档

- 反馈子系统 `message-feedback` 的类型、服务契约与 Web 消费方。
- 会话通过 `/feedback` 确认文本披露的共享策略。
- 匿名用户通过反馈确认文本的按 harness home 共享 id。

开发备注

无。

identity/ — 共享身份

---

description: "identity 包组：由遥测、反馈与 DeepSeek 提供方请求共享的匿名按 harness home 关联 id。"

kind: "package-group"

English | 中文

概述

identity 组为每个 harness home 提供一个匿名 id，该安装的遥测、反馈与 DeepSeek 请求会把它附加到各自的记录上，因此离开同一个 home 的所有内容都能被识别为来

目录

- 包
- 相关文档
- 开发备注

包

包	职责
anonymous-user-id	让每个 harness home 拥有一个匿名 id，遥测、反馈与 DeepSeek 请求把它附加到记录上，使来自同

相关文档

- [illegible]在系统中携带该 id 的遥测功能。
- dsh-llm-deepseek——在请求中携带该 id 的 DeepSeek 提供方。

- dsh-command-feedback——□□□□□□□□□□□□□□□□□□□□

□□□

□□

llm/ — LLM □□□□

---
description: "LLM □□□□□□□□□□□□□□□□□□□□DeepSeek □ pi-ai □□□□□□□□□□□□□□□□□□□□□□□□□□□□ token □□□"
kind: "package-group"

English | □□

□□

llm □□□ harnes llm □□□□□□□□□□□□ llm-retry □□□□□□ age token-meter □□□□□□□□□□□□□□□□□□□□

包	职责	ctx key
token-meter/	用固定启发式规则从持久会话日志测量请求与上下文压力	ctx.tokenMeter

相关文档

- LLM 流式子 StreamChunk 请求与适配器约定。
- Token 计量子 ctx.tokenMeter 背后的测量语义。
- 孪生 LLM 适配器什么 DeepSeek 路由交付两个结构不同的适配器。
- 按路由的 IO 下文如何路由模型请求并压缩上下文。
- 回放 token 计量孪生回放感知的计量背后的设计。

开发备注

无。

e2b/ E2B 远程运行时提供方

---

description: "E2B 远程运行时组映射：把文件与命令工作放进一个远程 Linux 沙箱，供 E2B 家族的用户与维护者浏览。"

kind: "package-group"

English | 中文

概述

e2b 组把 agent（智能体）的文件与命令工作移入远程 Linux 沙箱：文件读写、shell 命令与终端都在同一个远程世界中运行，而不是在你的机器上。三个包协同工作——一个提供共享沙箱，一个

目录

- 包
- 相关文档
- 开发备注

包

包（**package**）	职责	**ctx** 键
e2b	文件与命令工作运行所在的共享远程 Linux 沙箱	ctx.e2b
fs-e2b	远程沙箱内的文件读取、写入、编辑与列表	ctx.fs
subprocess-e2b	远程沙箱内的 shell 命令与交互式终端	ctx.subprocess

相关文档

- 可移植执行世界如何可以在不移动 harness 的情况下迁移，以及哪些内容留在本地。
- 子进程子系统 seam 约定与生成 `ctx.e2b` 表面，包括
- 文件系统文件系统 seam 约定与生成的 Cordis 表面。

开发备注

维护者的工作上下文——点击展开

无。

subprocess/: 子进程能力家族

---
description: "subprocess 组地图: 共享的子进程服务及其本地宿主提供方，供浏览本组的用户与维护者阅读。"
kind: "package-group"

English | 中文

概述

harness 运行的每个子进程与终端会话——bash 命令、语言服务 ctx.subprocess sub 观 gen，后端 [illegible] 执行。它不是独立的产品功能：消费方能力 sea

目录

- 包
- 相关文档
- 开发备注

包

包	职责	**ctx** 键
subprocess	定义子进程服务：可执行文件查找、受管进程 spawn 与真实终端会话	ctx.subprocess
subprocess-local	在宿主机器上运行这些进程与终端 spawn	[ctx.subprocess
win32-process	归属受限进程创建、stdio 、Job 分配、等待与句柄清理所用的共享 Win32	绑定库，不使用 ctx key

即使消费方重载，进程生命周期仍由服务负责管理；消费方负责定义进程的含义（一条 bash 命令、一个语言服务器），以及决定塑造该进程的每一项默认值。

相关文档

- [redacted]spawn spec[DSH_*结果]受管的
- subprocess seam Agent Note——bash 执行器的进程部分为何成为独立的 seam。

开发备注

维护者的工作上下文——点击展开

无。

shell/ — bash 能力家族

---

description: "面向部署方与维护者的 bash 能力家族说明，用于选择并组合 shell 执行器、沙箱化与面向模型的 bash 与 pwsh 工具。"

kind: "package-group"

English | 中文

概述

shell 组为 agent 提供命令执行能力：运行前台命令并读取其有界输出，或启动后台进程并轮询它—— bash 和 pwsh 工具位于所在 Windows、POSIX 上下文选择 Bash 执行器每 Wi

目录

- 包
- 相关文档
- 开发备注

包

包	职责	ctx key
shell	定义执行器约定：前台运行、后台句柄与请求解析	ctx.shell
bash-local	在 POSIX bash -c 进程运行 Bash 命令	[ctx.shell
bash-sandbox	通过沙箱能力限制 Bash 命令运行，并把拒绝报告为事实	[ctx.shell
pwsh-local	在 Windows pwsh -Command 进程运行 PowerShe 命令	[ctx.shell
pwsh-sandbox	通过沙箱能力限制 PowerShell 命令运行	[ctx.shell
shell-env	提供每条 sh DSH_* 环境的受管	ctx.shellEnv

包	职责	ctx key
tool-bash	[bash 工具向模型公开 Bash 执行与后台任务	[ctx.tools
tool-bash-persistent	在单个限定所有者范围的持久 Bash 会话中运行模型的 shell 调用	[ctx.tools
tool-pwsh	[pwsh 工具向模型公开 PowerShell 执行	[ctx.tools
tool-pwsh-persistent	在单个限定所有者范围的持久 PowerShell 会话中运行模型的 shell [用	[ctx.tools

profile 层恰好选择一个执行器实现（win32 层[ctx.sandbox][ctx.sandboxPolicy]务[illegible]册[illegible]拥有[illegible]面向模型工具。沙箱化组合还会选择一个

相关文档

- Bash 执[illegible]器[illegible]系[illegible]享的请求/spec 词汇、结果、后台进程与完整的服务约定。
- [illegible]绕沙箱执行器所消费的隔离能力。

开发备注

None.

terminal/□□□ PTY □□□□

---
description: "□□□□□□□□□□□□□□□□□□□□□□ ctx.terminals □□□□□□□□ bash □ pwsh □ shell □□□□□ 6 □□□□□□□□□"
kind: "package-group"

English | □□

□□

terminal/ □□ agent □□□□□□□□□□□□□□□□□□she terminal/ □□ ctx.terminals □□□□□□□□□□□□ terminal-bash/ +□□□□□□□□□□□□□□□□□□□

相关文档

先从子系统参考了解共享类型与服务接口面，再从 Agent Note 了解设计理由与暂缓边界。

- 用于 id 与 `ctx.terminals` 的 API 取，以及生成的
- 持久 PTY Agent Note——设计决策、备选方案与延期工作。
- 能力 seam——本家族遵循的 Service Definition / Service Provider / Consumer 拆分。

开发备注

维护者的工作上下文——点击展开

无。

code-runtime/——代码执行能力族

---
description: "代码执行能力族的包映射：程序执行能为你做什么，以及每个部分由哪个包负责。"
kind: "package-group"

English | 中文

概述

code-runtime/ ctx.codeRuntime [程序] 当作 [运行] 在 [] 中执行，返回 [] 和 [] 的 [] 包 [] host 与 CPython Python 后端服务。每次运行彼此独立——程序之间不保留任何状态——失败也会作为结果的一部分返回，调用方因此能知道程序为何失败，并把它反馈给模型。

目录

- 包
- 相关文档
- 开发备注

包

这三个包共同提供程序执行能力；每个 README 描述其各自部分做什么。

包	角色	ctx 键
code-runtime/	定义代码运行时做什么：针对宿主提供的绑定运行一个程序，并报告其打印和返回的内容	ctx.codeRuntime
code-runtime-worker-thread/	在全新的 Node Worker 线程中执行 TypeScript 程序	[ctx.codeRuntime
experimental/code-runtime-python/	实验性 Python 后端：持有 Node host 与 CPython 子进程之一的 fd-3 协议格式与 CPython 运行时实现	

相关文档

先从子系统参考了解服务约定，再看消费此能力的 PTC mode 设计，以及它所遵循的能力 seam 模型。

- [illegible]`ctx.codeRuntime` 的 cordis 接口面。
- PTC mode Agent Note——`run_code` 呈现给模型。
- 能力 seam——本家族遵循的 Service Definition / Service Provider / Consumer 拆分。

开发备注

维护者的工作上下文——点击展开

无。

sandbox/ 进程限制 seam; bwrap/Landlock/Seatbelt 后端

---
description: "进程沙箱包组：隔离 seam、各平台后端、共享策略解析器与 Windows 写入限制档。"
kind: "package-group"

English | 中文

概述

sandbox/ [read-only | workspace-write | danger-full-access | sandbox/)办: 隔离 Linux、macOS 与 Win sandbox-local/）、[sandbox-policy/）与 Window sandbox-windows-acl/）。被策略拒绝的受限调用可以通过用户批准的一次性升权重试。隔离仅限同世界：它与宿主共享内核与文件系统，容器、m

目录

- 包
- 相关文档
- 开发备注

包

四个包承担隔离角色；子系统参考文档拥有穷尽式约定与逐调用策略语义。

包	职责	ctx key

sandbox/	隔离服务约定：模式、强制执行、逐调用策略与升权词汇	ctx.sandbox
sandbox-local/	各平台隔离后端：Linux bwrap 与 Landlock、macOS Seatbelt、Windows 受限令牌	[ctx.sandbox
sandbox-policy/	共享策略归属：每个执行家族的部署默认值与逐会话模式覆盖	ctx.sandboxPolicy
sandbox-windows-acl/	Windows 写入限制：受限子进程只能写入工作区与私有临时目录	—[sandbox-local 挂载为 win32 后端）

相关文档

先从子系统参考文档了解共享词汇，再看隔离决策及其跨家族扩展。

- 进程沙箱程序系统逐调用策略、包装 argv 方言与故障关闭错误。
- 子进程沙箱边界、升权编排与延期阶段。
- 跨家族文件沙箱共享策略归属与沙箱化文件系统提供方。
- Windows ACL 受限令牌沙箱选择原始 ACL 受限令牌而非 mxc 与 AppContainer。

开发备注

维护者的工作上下文——点击展开

无。

fs/ 文件系统能力系列: seam、本地实现、面向模型的

---
description: "文件系统包组: `ctx.fs` 提供方约定、本地与沙箱强制后端、编辑前读取策略插件，以及面向模型的文件与搜索工具。"
kind: "package-group"

English | 中文

概述

fs/ 组为 age fs/ 智 ctx.fs fs-local/ fs-sandbox/ fs-observation-policy/ tool-fs/ read read_image write edit tool-fs-search/ glob grep ）提供面向模型的工具。部署挂载一个后端，加载策略以获得新鲜度防护的变更，并注册模型应看到的工具包；后端可以更换，无需改动工具或策略。文件 I/O 有意不设超时: de

目录

- 包
- 相关文档
- 开发备注

包

fs-e2b 同族把文件系统角色；子系统参考文档拥有穷尽式约定与错误分类体系。

包	职责	ctx 键
fs/	ctx.fs 服务约定：执行世界路径、有界文本 I/O，以及带可选版本防护的原子变	ctx.fs
fs-local/	宿主文件系统后端：读取、写入并编辑本机上的真实文件	ctx.fs
fs-sandbox/	沙箱强制后端：按每次调用的沙箱模式约束写入与编辑，读取直接通过	ctx.fs
e2b/fs-e2b	以 E2B 为后端：文件状态位于与 E2B 子进程提供方共享的远程执行世界	ctx.fs
fs-observation-policy/	fs/* 策略	fs/* 监听器

□	□□	**ctx □**
tool-fs/	[read read_image write [edit □□□□□□□	[ctx.tools
tool-fs-search/	□□□ ripgre glob][grep □□□□	[ctx.tools
tool-str-replace-editor/	[str_replace_editor [ctx.fs [view create □ str_replace [insert	[ctx.tools

fs-sandbox [tool-fs-search]□□□□□□

lsp/□□□□□□□□□

---
description: "lsp □□□□□□ LSP seam□□ stdio □□□□□□□□□ lsp □□□□□□□□□□

相关文档

- LSP 导航子系 `LspError` 以及。
- LSP 能力 seam Agent Note —设计原理、备选方案与刻意推迟的 API。
- 生成的工 `lsp` schema。

开发备注

维护者的工作上下文——点击展开

无。

skill/ — skill□□□□□□□

---
description: "skill □□□□□□□□□□□□□□□□□ skill □□□□□□□□ agent □□□□□□□□□□□□□□□□□□"
kind: "package-group"

English | □□

□□

skill □□ agent□□□□□□□□□□□□□□□□□□□□ skill □□□□skill□□□□ /name □□□□□□□□□□□□□□□□□□□□

相关文档

先从子系统参考了解共享词汇，再阅读 Agent Note 了解设计依据。

- skill 注册表、提供方约定、本地发现优先级，以及目录与工具。
- skill 系统 Agent Note 家族如何拆分与分层注册表设计。
- skill 目录热刷新 Agent Note 的初始目录与替换生命周期。
- skill 调用策略 Agent Note 模型与用户调用控制。

开发备注

维护者的工作上下文——点击展开

无。

compaction/ — □□□□□

---
description: "□□□□□□□□□□□□□□□□□□□□ /compact □□□□□□□□□□"
kind: "package-group"

English | □□

□□

compaction/ □□□□ agent □□□ /compact □□ dsh □□□□□□□□□□□□□□□□□□□□

相关文档

先从子系统参考了解共享词汇，再阅读两份 Agent Note 了解设计依据。

- 压缩能力 seam Agent Note 家族如何拆分，以及为何依赖会话与 LLM 词汇。
- 排队手动压缩 Agent Note /compact 如何与运行中的轮次串行化。
- 能力 seam —本家族遵循的 Service Definition / Service Provider / Consumer 拆分。

开发备注

维护者的工作上下文——点击展开

无。

context/ — 请求上下文插件

---
description: "context 组地图: 不定义工具、为每次请求添加持久且模型可见上下文的插件，供浏览本组的用户与维护者阅读。"
kind: "package-group"

English | 中文

概述

context @file mention 请求作用域的 agent-instructions dsh-base 默认包 mcp-profile patch 可以禁用）外，其余全部需主动启用。上

目录

- 包
- 相关文档
- 开发备注

包

包	职责	ctx key
agent-instructions/	AGENTS.md CLAUDE.md 工作区指令加载到上下文，并在文件编辑后刷新	
session-reference/	引用其他会话：提及一个会话，其有界只读快照即成为上下文	ctx.sessionReferenceResolver
file-reference/	@file mention 发现与供宿主驱动 UI 使用的共享 mention	ctx.fileReferences
file-reference-local/	@file mention 的本地工作区补全提供方	—
time-context/	每个步骤的当前时间、浏览器时区与经过时长	—
tmux-context/	agent 所在的 tmux session、window 与 pane 位置	—

□□□

- □□□□□mention URI□□□□□□□□□□□□□□□
- □□□□□□□□□□ agent□□□□□□□□□□□
- □□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□——□□□
□□

subagent/ subagent 能力系列：提供方[

---

description: "subagent 包组：委派 seam、其进程内与进程外后端，以及面向模型的委派工具。"

kind: "package-group"

English | 中文

概述

subagent 组是委派能力家族：它让 agent（智能体）把任务交[ctx.subagents 续派 agent 具的提供方并因此单个 agent 进程内子 agent 约定新启动，或从父级已完成历史派生。SDK 运行的完整 Harness 运行时。面向模型的工具向 agent 公开委派、相邻 Agent 消息与列举，父级总能看到存在哪些子级、它们在线还是仅存于存储。本页是组的映射；各包 README

目录

- 包
- 相关文档
- 开发备注

包

包	职责	**ctx** 键
subagent/	定义委派服务：提供方注册表、一次性运行、可继续子级与发现	ctx.subagents
subagent-in-process-driver/	提供共享的进程内运行驱动器	无
subagent-spawn-in-process/	运行全新的进程内子 agent	[ctx.subagents
subagent-fork-in-process/	运行从父级已完成历史派生的进程内子 agent	[ctx.subagents
subagent-acp/	经 Agent Client Protocol 运行进程外子 agent	[ctx.subagents
subagent-codex/	经官方 app-server 协议运行真实 Codex 子 agent	[ctx.subagents

包	职责	**ctx** 键
subagent-claude-code/	经官方 Agent SDK 运行真实 Claude Code 子 agent	[ctx.subagents
subagent-dsh-sdk/	经 TypeScript SDK 运行进程外 Harness 子 agent	[ctx.subagents
tool-subagent/	向模型公开委派	[ctx.tools
tool-subagent-control/	向模型公开相邻 Agent 消息、中断与列举	[ctx.tools

相关文档

- Subagent □□□ 服务约定、提供方约定与终态结果语义。
- Subagent 能力 seam —— 委派能力家族的设计记录。
- 可续跑后台 subagent[受后续轮次的持久子级。
- 合并后的 subagent 控制服务、中断与列举面。

开发备注

无。

jobs/: 后台任务能力家族

---

description: "jobs 组地图: 后台任务控制——注册表约定、进程本地存储与面向模型的任务工具，供浏览本组的用户与维护者阅读。"

kind: "package-group"

English | 中文

概述

jobs 组是后台工作能力家族: 运行长时间工作的工具把工作注册为任务，拥有它的 `jobs` `jobs-local` `tool-jobs` 模型侧控制面取消任务。任务属于启动它的 agent 会话，因此一个 agent

目录

- 包
- 相关文档
- 开发备注

包

包	职责	**ctx** 键
jobs	定义后台任务约定: id、归属、生命周期与完成监听器	ctx.jobs
jobs-local	在本进程中运行并存储任务，按所有者隔离	[ctx.jobs
tool-jobs	让模型读取、列出和终止任务，并投递完成通知	[ctx.tools

相关文档

- 使用 `ctx.jobs` 的 API。
- 通用长时间运行工具运行时 Agent 任务 Note 后的设计。
- 任务注册表 seam Agent Note 所有者隔离的注册表约定及其理由。

开发备注

维护者的工作上下文——点击展开

无。

experimental/ 私有原型与内部专用插件

---

description: "实验组地图：不进入正式发布的私有原型与内部专用插件，供浏览本组的用户与维护者阅读。"

kind: "package-group"

English | 中文

概述

实验组包含不属于任何正式发布的原型能力：它们运行在真实 harness 上，但约定可能变更，也不提供支持承诺。本组包含 Agent Teams、跨 realm Inspector、代码执行 sea

目录

- 包
- 相关文档
- 开发备注

包

包	职责	ctx 键
agent-team-profile	Agent Teams 的显式源码 checkout profile 层	—
agent-team	具名 teammate，成员之间持久消息与共享任务板	ctx.agentTeams
agent-team-web-profile	Agent Teams 的显式源码 checkout Web 层	—
client-ui-agent-team	Web Team roster、任务板与 teammate 导航	—
code-runtime-python	代码执行 seam 的 CPython 子进程后端	ctx.codeRuntime
inspector	用于 Host 调试、Client Runtime 检查、网络采集与 Cordis CDP hub	ctx.inspector

包	职责	**ctx** 键
tool-agent-team	让模型创建、发消息与协调 teammate 的十个工具	[ctx.tools
webworker-packer	构建浏览器 worker 预览所消费的 gzip 压缩 VFS 镜像	库与 CLI，不使用 ctx key
webworker-runtime	在专用浏览器 worker 中运行 harness 插件树	库与 worker 入口，不使用 ctx key

相关文档

- 实验包决策、配置、发布排除与依赖隔离。
- Agent Teams 子系统—持久 Team ctx.agentTeams 服务 API。
- 实验子树如何跨越状态放宽了什么、不放宽什么。

开发备注

维护者的工作上下文——点击展开

无。

workflow/ □□□ seam□worker □□ workflow/ralph □□

---
description: "workflow □□□□□□□□□□□□□□ subagent □□□□□□□□□□□□□□□□□□□□□□□□"
kind: "package-group"

English | □□

□□

workflow □□ agent □□□□□□□□□ workflow □□□□□□□□□ subagent □ workflow □□ ralph □□□□□□□ agent □□□□□□□□□□□□□□□ agent□

□□

- □
- □□□□
- □□□□

□

□	□□	ctx □
workflow	□□□□□□□□□□ subagent □□□□□	ctx.workflowEngine
workflow-worker-thread	□□□ worker thread □□□□□□□□□□□□□□□□□□□□	□ ctx.workflowEngine
tool-workflow	□ workflow □□□□□□□□□□□□□ agent □□	□ ctx.tools
tool-ralph	□ ralph □□□□□□□□□□□ agent □□□□	□ ctx.tools

相关文档

- 工作流子seam [`workflow/*` 事件。
- 生成的工`workflow` 工具 schema。
- 生成的工`ralph` 工具 schema。
- 生成的配置自动支持的引擎配置字段。
- 动态工作流 Agent Notseam 设计及其决策。
- Ralph 工具 Agent Note—固定全新 agent 循环的设计与暂缓事项。

开发备注

维护者的工作上下文——点击展开

无。

webhook/ — 从已验证外部事件到 DSH Se

---
description: "经验证的外部事件、程序化规则与即发即弃 DSH Session 创建的包映射。"
kind: "package-group"

English | 中文

概述

Webhook 系列接收通过身份验证的提供方事件，并运行受信任的程序化规则。规则可以在 Web Workspace 中创建普通根 Session。分发仅存在于进程内并采用 fire-and-fc

目录

- 包
- 相关文档
- 开发备注

包

包	角色	ctx key
webhook/	规则注册表、回调生命周期与基于 Workspace 的 Session 创建	ctx.webhookRuntime
webhook-github/	签名 GitHub HTTP 适配器	[ctx.webhookRuntime [ctx.webServer

相关文档

null Session Method

web/: web 访问能力家族

---
description: "web 访问能力家族的包映射：搜索/抓取服务、其提供方后端，以及消费它们的面向模型工具。"
kind: "package-group"

English | 中文

概述

web/ 组为 harness 提供 web 访问能力——搜索 web（ctx.web）[illegible]，通过后端和提供方无关的服务接口一个或多个后端——搜索用 Exa、Perp[illegible] web/ 服务、[illegible] web_search [web_fetch [tool-web/。该组只拥有 web 访问本身：没有浏览或提取，没有逐 URL 策略，各后端保留自己的资源上限。搜索与抓取有意共用一项服务，使选择、取[illegible]

目录

- 包
- 相关文档
- 开发备注

包

六个包分别承担 web 角色；子系统参考文档拥有穷尽式词汇与约定。

包	职责	ctx 键
web/	搜索/抓取服务：通过可互换的后端搜索与抓取 URL，统一选择与错误策略	ctx.web
web-search-exa/	通过 Exa 搜索 web	[ctx.web
web-search-perplexity/	通过 Perplexity 搜索 web	[ctx.web
web-search-deepseek/	通过 DeepSeek 原生搜索搜索 web	[ctx.web
web-fetch-http/	匿名抓取公共 HTTP(S) 页面	[ctx.web

包	职责	**ctx** 键
tool-web/	[web_search [web_fetch	[ctx.tools

相关文档

先从子系统参考文档了解共享词汇，再看单一提供方选择服务背后的设计决策。

- web 工具 — [WebError 、[illegible]规则。
- web 能力 seam — 搜索与抓取为何共用一项提供方选择服务。

开发备注

维护者的工作上下文——点击展开

无。

attachment/□□□□□□□

---

description: "□□□□□□□

相关文档

先从子系统参考了解服务约定，再看能力 seam 表与本地后端的配置面。

- 附件子系 `ctx.attachments` 的 cordis 接口面。
- 能力 seam——本家族遵循的 Service Definition / Service Provider / Consumer 拆分。
- 生成配置与本地后端的每个受支持字段。

开发备注

维护者的工作上下文——点击展开

无。

spill/□□□□ spill □□□

---

description: "□□□□ spill □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"

kind: "package-group"

English | □□

□□

spill/ □□□□□□□□□ spill/ □ spill-local/ □□ spill-policy/ □□□□□□□□□□□□ maxInlineBytes □□ dsh-output-retention □□□□□□□□□□□□□□

□□

- □
- □□□□
- □□□□

□

□□□□□□□ spill □□□□□□□□□□□□□□□□□□□□□□

□	□□	ctx □
spill/	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	ctx.spillStore
spill-local/	□ spill □□□□□□□□□□□□□□□	□ ctx.spillStore
spill-policy/	□□□□□□□□□□□□□□□□□□□□	□ ctx.tools

相关文档

先从子系统参考文档了解共享词汇，再看设计决策及其持久日志扩展。

- spill 模块 `SaveTextSpill` / `SpillRef` 词汇、归属与后端关系。
- 工具输出 spill 决策、保留与工具自有输出处理之间的能力边界。
- 代码 dispatch-log spill 模块 `run_code` 子调用结果的持久副本同样设界。

开发备注

维护者的工作上下文——点击展开

无。

todo/ □□□□ todo_write □□

---
description: "todo □□□□□□□□□□□□□ todo_write □□□□□□□□□□□□□□□□□□□"
kind: "package-group"

English | □□

□□

todo □□ agen todo_write □□□□□□□□□□□□□ agent □ □□□□□□□□□□□□□□□□□□□

- □□□□[todo_write schema。
- □□□□□□□□□支持配置字段。
- todo_write 工具 Agent Note——原始设计及其备选方案。

开发备注

维护者的工作上下文——点击展开

无。

plan/□plan □□□□

---
description: "plan □□□□□□□□ agent □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"
kind: "package-group"

English | □□

□□

plan/ □□□□□□□ /plan □□agent□□ plan-mode □□agent□ □□□□□□□□□□□□□□□□□□□□

相关文档

先从子系统参考了解共享词汇，再阅读设计说明了解决策。

- 计划模式子系统参考 — 规划模式如何工作、其配置与退出工具的行为。
- plan 专用协作状态 — 规划模式背后的设计决策。

开发备注

维护者的工作上下文——点击展开

无。

preset/ 由 preset cordis.yml 按 agent

---

description: "preset 组地图: 按会话从 preset 文件组装 agent, 供浏览本组的用户与维护者阅读。"

kind: "package-group"

English | 中文

概述

preset 组提供按会话的 agent（智能体）组装 agent.cordis.yml；你 preset 组装的会话会运行该 agent-presets 拥有者；skill 对话配置根目录会 harness agent 挂载，persona 式则已供已组装的行，让 preset 不止能改变 agent 的工具、也能改变它的身份。两者合起来让一个进程可以同时运行多个组装方式不同的 agent。

目录

- 包
- 相关文档
- 开发备注

包

包	职责	ctx 键
agent-presets	preset 名单、对受信任根目录与用户根目录的发现、按 agent 组装、只复制	ctx.agentPresets
persona	preset 挂载的可组装人设行，用于遮蔽或替换部署级人设	—

相关文档

- AgentPresets 参考——发现、挂载、继承与重组。
- Scope 子系统–scope key 与挂载用以加入 agent 的父链。
- 系统提示词reset 提示词段落如何注册与组装。
- 按会话组装 agent preset 的 Agent Note 理由与备选方案。
- 按 preset 常驻挂载的 Agent Note 为何是常驻且共享的。

部署交付的 pre agent-presets/presets/ ———个 preset ——个目录，那份目录列表就是名单；在这里再列一遍只会多出一份需要同步的名单。

开发备注

维护者的工作上下文——点击展开

无。

guard/: 循环卫生 guard 家族

---

description: "循环卫生 guard 家族的包映射：建议性重复工具提醒与单次工具调用超时策略，供选择或组合 guard 的用户与维护者阅读。"

kind: "package-group"

English | 中文

概述

guard/ 组通过监视两种常见失败[illegible] repeat-tool-reminder [illegible] timeout-policy [illegible] dsh [illegible] 清晰的超时错误，而不是拖住

目录

- 包
- 相关文档
- 开发备注

包

两个小插件分别覆盖两种模式；下文每个 README 都说明何时保留、调优或移除它。

包	提供什么
repeat-tool-reminder/	在模型重复相同工具调用时提醒它，使其改变方法或结束任务
timeout-policy/	为声明了限时的工具调用设置超时，让模型得到清晰错误而不是无限等待

相关文档

先从工具子系统参考了解工具调用流水线，再看重复提醒的配置与策略背后的超时库决策。

- 工具子系统参考 Guard 都依赖的工具调用流水线与决策。
- 生成配置参考 重复调用提醒的每个受支持字段。
- 超时截止时间库 Agent N `timeout-policy` 所执行的时序／终止拆分。

开发备注

维护者的工作上下文——点击展开

无。

bundle/: profile 插件组合包

---
description: "共享核心、浏览器 GUI、一次性任务、ACP 与 SDK 应用表层的现成 dsh profile bundle。"
kind: "package-group"

English | 中文

概述

[dsh --profile 使用的可安装 p dsh.bundle.patch；启动器会叠放这些 patc web headless acp [sdk profile [dsh-base sdk-minimal 则由一个 bundle 提供完整配置树。领域包也可以在本目录之外声明附加层。

目录

- 包
- 相关文档
- 开发备注

包

包	职责	ctx key
base	基于 base 的 profile 共享核心	—（仅 patch）
acp-app	基于 base 的纯自动化 ACP stdio 应用	挂载 ACP bridge
web-app	基于 base 的浏览器应用层	挂载 Web 配置项
headless	基于 base 的一次性命令行任务应用	headless-runner
sdk-app	基于 base 的 SDK JSON-RPC stdio 应用	挂载 SDK server
sdk-minimal	不使用 base 或 Web 的独立极简 SDK 应用	—（完整 patch 树）

内置组合包从 dsh 安装目录解 `dsh plugin --profile <name> add <package>` 安装进 profile。

相关文档

- dsh 应用—启动 profile `dsh` 命令。
- app-boot——profile 如何解析、分层与定制。
- Profile 组合包设计 profile 与组合包的组合设计。
- 生成组合包个已发布 profile 使用的确切组合。

开发备注

无。

extensions/ agent 运行时自修改：实时插[

---

description: "extensions 组地图：用于定义、运行与移除动态 Cordis 包的模型侧工具和双半 runner，供浏览本组的用户与维护者阅读。"

kind: "package-group"

English | 中文

概述

extensions 组让运行中的 agent 修改它自己所在的运行时：模型可以检查当前 DSH 进程里加载的插件与服务，定义动态 Cordis 包（可含 host 半、浏览器半或两者），运行、停止并

目录

- 包
- 相关文档
- 开发备注

包

包	职责	**ctx** 键
tool-cordis	七个模型侧工具：检查实时运行时，定义、运行、停止并移除动态包	[ctx.tools
cordis-host-runner	host 半：定义注册表、沙箱化的 host 半生命周期，以及浏览器查询应答的 ins[	ctx.dynamicCordisRunner 与 ctx.cordisInspect
cordis-client-runner	浏览器半：把浏览器半源码求值成活插件，并应答运行请求	client [ctx.dynamicCordisRunner
ui-cordis	[@pluginId]命令调用工具卡片与	client 面；注册 slot

相关文档

- extensions 子系统–[ctx.cordisInspect [ctx.dynamicCordisRunner 服务 API。
- 生成的工具的模型侧工具 schema。
- 生成的配runner 的受支持配置字段。
- 自引用 Cordis 工具集 Agent Note 语义、生命周期与组合的设计居所。
- 客户端外壳与动态包 Agent Note 包归属与构建面。

开发备注

维护者的工作上下文——点击展开

[packages/client/ 下，因为它们是本子系统双半包的其中一半；client 面经由 client program 编译它们，host program 只引用 host runner。

hooks/ □□□□ + □□□ Claude Cod Codex □□□□

---
description: "hooks □□□□□ agent □□□□□□□□□ Claude Code □ Codex shell □□□□□□□□□□□□□□□□□□□□□□"
kind: "package-group"

English | □□

□□

hooks □□ agent□□□□□□□□□□□□□ Claude Code hooks.json □□□□ shell □ agent□□□□□□□□□□□□□□□□□□□

相关文档

- 拦截扩展点 Agent Note 接所面向的类型化 Decision 接口面。
- 钩子桥接 Agent Note 桥接设计及其决策映射。
- 钩子协议库 Agent Note 享库负责的内容及其原因。

开发备注

维护者的工作上下文——点击展开

无。

session/ — □□□□□□□

---

description: "□□□□□□□□□□□□□□□ seam □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"

kind: "package-group"

English | □□

□□

session □□ agent□□□□□□□□□□□ loop □□□□□□□□□□□ seam □□ session-query/ □□□□□□□□□□□□□□□□□□□□

包	职责	ctx key
session-log-deepseek/	把增量规范日志作为可选的官方 DeepSeek 请求元数据上传	[dsh_session_log

投影

包	职责	ctx key
session-projection/	定义并驱动把已提交事件折叠为完整当前值的投影单元	ctx.sessionProjections
session-projection-cache/	持久化投影检查点，使冷读跳过全量日志加载	ctx.sessionProjectionCache
session-stats/	[sessionStats 单元提供全日志会话计数与墙钟时间	[ctx.sessionProjections
session-turn-outline/	[turnOutline turn/start [illegible]，带示词预览）	[ctx.sessionProjections

标题

包	职责	ctx key
session-title/	基于日志的会话标题，带确定性回退与一个可选提供方	ctx.sessionTitle
session-title-llm/	供提供方包共享的模型标题生成策略	库，不使用 ctx key
session-title-first-prompt-llm/	根据第一条合格的人类消息为会话生成标题	[ctx.sessionTitle
session-title-all-prompts-llm/	根据所有合格的人类消息为会话生成标题	[ctx.sessionTitle

遥测

包	职责	ctx key
session-telemetry/	捕获会话活动并把记录交给配置的上报后端	ctx.sessionTelemetry
session-telemetry-otel/	通过 OpenTelemetry FULL FEEDBACK_ONLY [DISABLED 模式投递遥测	[ctx.sessionTelemetry

同一时间只允许一个标题提供方注册；未注册时，标题服务保留其确定性回退。下面的子系统页面是各家族后端无关的参考资料。

相关文档

- [illegible]后端系统的服务语义、flush 检查点与崩溃恢复。
- [illegible]投影单元约定与驱动语义。

- 会话标题存取器资格、回退与提供方流程。
- 会话遥测捕获系统脱敏与投递模式。
- 会话子系统每个包持久化或派生的实时事件日志。

开发备注

无。

session-query/□□□□□□□□

---
description: "□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Web □□□□□□□□"
kind: "package-group"

English | □□

□□

session-query/ □□□□□□□□□□□□□□□□□□□□□□□□comp /export □□□□□□□□□□□□□□□□□□□□

相关文档

先从子系统参考了解共享的查询词汇，再看追踪、搜索与面向模型工具背后的设计记录。

- 会话查询逻辑记录过滤器、搜索页、血缘、有界读取与事件关系。
- 会话查询追踪与校验边界。
- SQLite FTS5 会话搜索搜索语义、对账与 tokenizer 决策。
- 面向模型的会话概览与工具游标结果设计。

开发备注

无。

settings/: 用户可编辑配置

---

description: "用户设置能力族的包映射：解析各 namespace 配置的 ctx.settings 服务，以及存储它的 YAML/JSON 文件提供方。"

kind: "package-group"

English | 中文

概述

settings/ 组让插件配置变为用户可编辑：插件用一个 sc cordis.yml [illegible] settings/ [illegible] settings-file/ [illegible] 存进一个用户可编辑的 YAI

目录

- 包
- 相关文档
- 开发备注

包

两个包覆盖该能力；完整约定由各子级 README 负责，穷尽式服务接口面由子系统参考负责。

包	角色	**ctx** 键
settings/	设置服务：注册 namespace 并读取或修改其值	ctx.settings
settings-file/	把设置存进一个本地 YAML/JSON 文件并热发布外部编辑	[ctx.settings

相关文档

先从子系统参考了解共享词汇，再看本家族遵循的能力 seam 拆分。

- □□□□□□namespace、分层解析、descriptor、变更提交与生成的 cordis 接口面。
- 能力 seam——本家族遵循的 Service Definition / Service Provider / Consumer 拆分。

开发备注

维护者的工作上下文——点击展开

无。

credentials/: 凭据与授权

---

description: "凭据能力族的包映射：凭据引用 seam、环境与文件提供方、授权 flow 注册表，以及引用如何让机密值留在配置之外。"

kind: "package-group"

English | 中文

概述

credentials/ 组管理你的配置按名引用的机[cordis.yml][credentials/ credentials-local/][责存储与][authorization/ ）——用于获取无法配置、只能开口去[DEEPSEEK_API_KEY=… dsh ）始终优先于存储值。机密值绝不进入你同步或渲染的配置文件——进去的只有它们的名字，而且本地文件只有同– OS 用户可读，其他用户读不到。

目录

- 包
- 相关文档
- 开发备注

包

三个包共同提供凭据功能：一个在运行时存储、查询与移除机密，而配置只写名字；第二个是默认的本机存储；第三个让插件获取必须开口去要的凭据。它们的 README 覆盖日常使用；子系统参考拥有穷尽式约定。

包	角色	**ctx** 键
credentials/	在运行时存储、查询与移除机密，而配置只写名字	ctx.credentials
credentials-local/	默认本机存储：一个私有 YAML 文件，环境覆盖优先	[ctx.credentials
authorization/	由插件拥有、通过询问人来取得凭据的 flow	ctx.authorization

相关文档

先从子系统参考了解共享词汇，再看能力 seam 表与本地存储的配置面。

- 用于 `CredentialRef`（`CredentialKey`、按操作 `CredentialInfo`、授权 flow 与生成的 cordis 接口面。
- 能力 seam——本家族遵循的 Service Definition / Service Provider / Consumer 拆分。
- 用于本地存储的每个受支持字段。

开发备注

维护者的工作上下文——点击展开

无。

storage/ 非会话存储中枢 + 后端 + 领域形式

---

description: "存储组地图：通过具名后端与类型化领域数据形式持久化非会话数据，供浏览本组的用户与维护者阅读。"

kind: "package-group"

English | 中文

概述

存储组为组合提供会话事件日志以外一切数据的持久存储：工作区记录、会话伴随数据，以及其他宿主侧应用数据。借助它，宿主包可以经 schema 校验过的领域数据形式持久化类型化记录，在人类可读的 JSON

目录

- 包
- 相关文档
- 开发备注

包

包	职责	**ctx** 键
storage	把已注册后端与已挂载的数据形式设施连接起来	ctx.storage
storage-json	把每个单元存为一个人类可读的 JSON 文件	[json
storage-sqlite	把单元作为 JSON 文档存进一个 SQLite 数据库	[sqlite
storage-domain	在已路由后端之上提供经过 schema 校验、发出变更事件的 KV 领域	ctx.storageDomain

---

□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ API□
- □□ KV □□ Agent Note□□□□□□□workspace □□□□□□□□□□□□□□□□
- Workspace □□□—□□□□□□□□□□□□□□□

□□□□

□□□□□□□□—□□□□

□□ Agent Note □□□□ pro log ed□□□□□□□□□□□□□□□□□□□

workspace/ Workspace 实体

---

description: "workspace 组地图：持久工作区实体家族、用户目录的持久记录与基于会话头的成员资格记账，供浏览本组的用户与维护者阅读。"

kind: "package-group"

English | 中文

概述

workspace 组提供 workspace 目录的持久记录，用户可以在其中创建项目、保持稳定顺序，并把每个项目的会话归入其下。借助它，UI 可以显示带会话的项目侧边栏、把会话从分组中隐藏而不删除它，以及移除项目

目录

- 包
- 相关文档
- 开发备注

包

包	职责	**ctx** 键
workspace	提供命名且有序的项目，并在每个目录下聚合其会话	ctx.workspaceRegistry

相关文档

- Workspace 项目及其会话的权威功能约定。

- 领域 KV 存储 Agent Note 项目记录背后的存储设计。
- Workspace UI 产品流 Agent Note 首次启动如何从会话历史构建项目，以及 GUI 如何排序。
- 删除 Workspace 时是否删除项目绝不会删除其文件夹或会话。

开发备注

维护者的工作上下文——点击展开

无。

sdk/: 从另一进程驱动 Harness 运行时

---
description: "SDK 能力家族的包映射: JSON-RPC 协议格式，以及供进程外 SDK 使用的 TypeScript 客户端与服务器。"
kind: "package-group"

English | 中文

概述

本组让另一进程驱动完整的 DeepSeek Harness 运行时: JSON-RPC 协议格式定义消息，服务插件通过 stdio 为外部 `dsh` 服务组[illegible]完成事件。TypeScript 客户[illegible]是 SDK 的设计孪生，二者说同一种协议。本页是组的映射；各包 README 负责各自的包级约定。

目录

- 包
- 相关文档
- 开发备注

包

每个包 README 描述你可用该包部分完成的事情。

包	职责
protocol/	协议格式: 按换行分帧的 JSON-RPC 传输，以及具名的请求、结果与通知类型
client/	TypeScript 客户端: 启动运行时子进程，通过高层与协议层 API 驱动 agent 轮次
server/	`jsonrpc` 插件: 通过 stdio 为进程外 SDK 客户端提供服务

相关文档

先从 Python SDK（客户端约定的姊妹实现）开始，再看可运行应用与组边界背后的决策记录。

- Python SDK — 说同一种协议的 Python 对侧实现，并随附捆绑运行时。
- SDK 应用组合创建 JSON-RPC `dsh --profile sdk` 应用。
- Python profile 运行时消费后的 Python 客户端为何启动相同的具名 profile。
- TypeScript SDK 与 SDK subagent 一端客户端约定及其上的 subagent 后端。
- SDK 项目工具链脚手架从不创建、配置或构建开发者项目。
- SDK subagent 提供方harness 内部消费 TypeScript 客户端的例子。

开发备注

无。

acp/ — Agent Client Protocol

---
description: "ACP (Agent Client Protocol) 包组: 通过 JSON-RPC stdio 将全新 harness agent 暴露给程序化客户端的仅自动化服务器。"
kind: "package-group"

English | 中文

概述

acp 组提供一个包: 一台服务器，让程序与自动化可以通过标准 Agent Client Protocol 运行持久 DeepSeek Hari subagent/subagent-acp 。本项目的 README 负责各自的包级约定。

目录

- 包
- 相关文档
- 开发备注

包

包	职责
acp/	让程序通过 ACP 管理持久 agent、挂载 MCP 服务器、选择模型选项、发送或取消工作并接收语义更新

相关文档

开发备注

- dsh-subagent-acp——spawn □□□□□□□□□□ ACP □□□□
- ACP □□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□ACP□□□□□□□

□□□

□□

interaction/□□□□□□

---
description: "□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ agent □□□□□□□□□□ seam□"
kind: "package-group"

English | □□

□□

interaction/ □□□□□□□□□□□□□□□□ dsh □CLI□ □□□□□□□□□□□□□□□□□□□□

相关文档

先从子系统参考了解共享词汇，再看相邻的自动化与组合面。

- 命令子系统 —— `ctx.commands` 的 cordis 接口面。
- 审批子系统 —— 请求／结果词汇、应答者瀑布与按会话策略。
- 权限子系统 —— 预设、预授权与旋钮写穿。
- 用户交互子系统 —— 问题词汇、answerer waterfall 与呈现意图。
- ACP 组 —— 仅自动化的传输，为其自有 agent 回答审批请求。

开发备注

维护者的工作上下文——点击展开

无。

boot/□□□ app bin □□□□

---
description: "boot □□□dsh app bin □□□□——□□□□□profile □ patch □□□□□□□□□□□□□□□□□□□□□□□□□□□□"
kind: "package-group"

English | □□

□□

boot □□□□□ dsh app app-boot □□ cordis.yml □□□□□□□□ cmdline □□□□□□□□□ --help □ dsh □□□□□□□□□ apps/cli □□□□□□ fixture □□ fixture □□□□□□□□□□□□□□□□□□□□□□□□□□□□ README □□□□□□□□□□

□□

- □
- □□□□
- □□□□

□

□	□□	ctx □
app-boot	□ cordis.yml □□ dsh .env □□□ profile □ patch □□□□□□bin□□□□□	
cmdline	□□□□ --help □□□□□□□□□□ flag □□□□□□□□□	cmdlineArgs appExit

□□□□

- dsh 应用——在其启动序列中使 dsh helper 的
- Profile 组合包——[dsh --profile] 组合挂载的可安装 patch 层。
- dsh-home-paths——两个包都依赖的 harness home 解析器。
- 应用持有为待办决 flag 家族由应用持有而非启动器。

开发备注

无。

host/ — Web GUI 宿主侧

---
description: "Web GUI Host 侧的包映射: HTTP 与 SPA 服务器、工作区目录选择实现和插件清单投影。"
kind: "package-group"

English | 中文

概述

host/ 组提供 Web GUI 的普通 HTTP 服务器、服务已构建 Web 壳 client/ 的 apps/cli [dsh-base]组[apps/web/

目录

- 包
- 相关文档
- 开发备注

包

七个包分别承担 Host 角色; 各包的 README 拥有自己的约定与配置。

包	职责	ctx 键
webserver/	浏览器 HTTP 服务器: 具名路由、upgrade、index 转换与回退席位	ctx.webServer
frontend-static/	占据 webserver 回退席位的 SPA dist 服务器	[ctx.webServer
directory-picker/	工作区目录选择 seam: 能力约定与错误词汇	ctx.directoryPicker
directory-picker-native/	面向宿主屏幕前操作者的原生 OS 选择器后端	[ctx.directoryPicker
directory-picker-browse/	应用内目录浏览器后端，也服务于远程客户端	[ctx.directoryPicker

包	职责	ctx 键
directory-picker-auto/	在启动时挂载匹配后端的宿主自适应选择器	挂载一个后端
plugin-inventory/	当前 Loader 条目的只读投影	Remote `pluginInventory/list`

相关文档

先从传输与工作区记录的子系统参考读起，再看 Web Client 背后的分层决策。

- HTTP 服务器子webserver 的路由、匹配顺序与配置。
- 工作区子服务选择器所喂给的工作区记录。
- Web 配置树启W传T分层输各层的所有权。

开发备注

维护者的工作上下文——点击展开

无。

client/ — Web GUI 浏览器侧

---

description: "web GUI 浏览器侧的包映射：外壳启动、浏览器与宿主通信、共享客户端服务、本地化、开发重载与 UI 功能插件。"

kind: "package-group"

English | 中文

概述

client/ 组运行 dsh web GUI 的浏览器侧：它启动 web 外壳、加载浏览器侧插件模块、维持浏览器与宿主之间的 @deepseek-ai/dsh-client-<name> 对 host/ [AGENTS.md](), 模块图、slot 模型与对象层的说明见下方相关文档。

目录

- 包
- 相关文档
- 开发备注

包

内核包负责启动与服务于页面，UI 功能包负责呈现页面。各包的 README 拥有自己的约定与配置。

包	职责	ctx 键
web/	启动浏览器外壳	—
modules/	加载浏览器侧客户端模块	ctx.clientModules / ctx.modules
connection/	维护浏览器与宿主之间的 RPC 通信与事件投递	ctx.connection
store/	提供不依赖 React 的 observable 与 snapshot-store 七语	

包	职责	ctx 键
hmr/	在开发期间刷新客户端插件	—
locale/	提供本地化偏好与消息词典	ctx.locale
test-runtime/	为客户端功能包提供共享的仓库测试支持	—
ui-renderer/	将 slot 数据绑定到 React，并挂载组装完成的应用	ctx.uiRenderer
ui-slots/	定义 UI 功能注册与组合扩展 slot 的方式	—
ui-session/	把 Session Controller 状态适配为标准 Slot source 与 hook	
ui-theme/	应用所选颜色主题	—
ui-primitives/	提供共享 React 控件、图标与内容渲染器	—
ui-attachment/	注册输入框与消息图片的附件呈现	—
ui-layout/	排列应用的主要区域	—
ui-sidebar/	展示工作区与会话导航	—
ui-brand-official/	用官方名称与标记填充通用浏览器品牌 slot	—
ui-workspace/	提供工作区选择与创建界面	—
ui-conversation/	展示当前对话及其输入界面	—
ui-chat/	投影并渲染 Chat 对话 target	—
ui-approval/	展示批准请求并返回用户决策	—
ui-tool/	编排工具调用树与按工具键控的视图	—
ui-workflow-run/	把持久工作流运行回放为嵌套对话折叠项	—
ui-goal/	展示与管理当前目标	—
ui-trajectory/	提供 agent（智能体）活动的其他视图	—
ui-commands/	提供会话感知的命令发现与分发	—
ui-input-trigger/	协调内联命令与引用建议	—
ui-skill/	向内联建议添加 skill（技能）引用	—
ui-reference/	统一的 W @file / @session 引用 source	—
ui-subagent/	提供 subagent（子智能体）导航、子级 transcript（文本记录）状态与二联引用	
ui-schedule/	在只读标题栏目录中列出当前 Session 的活动提醒	—
ui-jobs/	在会话标题栏列出当前会话的后台任务	—
ui-model-selection/	在对话界面中提供模型选择	—
ui-permission-presets/	配置默认权限并切换当前会话的访问模式	—
ui-plan/	展示生效中的 plan mode 状态及其退出控件	—
ui-settings-plugins/	拥有“插件”设置分区、其标签页扩展点与可配置的宿主平面插件卡片	—
ui-user-questions/	展示 agent 请求的交互式问题	—

包	职责	ctx 键
ui-agent-preset/	选择会话的 agent 预设并编写预设组合	—
ui-settings/	承载设置界面及其扩展区域	—
ui-settings-general/	提供常规设置分区	—
ui-settings-models/	提供模型提供方配置与 DeepSeek 引导	—
ui-settings-plugin-inventory/	向“插件”设置贡献只读的 Host Loader 清单标签页	—
ui-deliverables/	生成已产出文件的轮次尾部与可点击的最终响应文件引用	—
ui-message-feedback/	向助手消息操作条贡献逐消息反馈控件	—
ui-directory-picker-browse/	面向工作区目录流程的应用内目录浏览界面	—
ui-directory-picker-native/	驱动宿主 OS 选择器的原生目录选择界面	—

相关文档

先从子系统参考与两份拥有跨包组合决策的 Agent Note 读起，再看服务于本页的宿主半侧。

- 客户端模块统 dsh.client 声明、启动图协议与 bundle 路由。
- slot 系统标准权威 slot 模型：注册、props 份额与 store。
- web 客户端架构 Agent Note 载链、对象层与客户端服务。
- 宿主组地服务于本浏览器半侧的宿主半侧。

开发备注

维护者的工作上下文——点击展开

无。

test-support/ 支持基础设施（testkit、不

---
description: "test-support 组地图：面向编写与运行仓库测试的开发者，提供无密钥测试工具、LLM mock 与回放服务器以及 Loader 冒烟测试辅助。"
kind: "package-group"

English | 中文

概述

test-support 组为仓库测试提供确定性、无密钥地运行真实产品的方式。它包含 Loader 应用 harness、session-log 快照适配器、回放 LLM（大语言模型）插件和可编脚本

目录

- 包
- 相关文档
- 开发备注

包

包	职责
session-snapshot	为 profile 驱动的测试提供 session-log 快照支持与协议适配器
agent-loop-testkit	为运行具体 AgentLoop 的测试提供共享先决服务
client-runtime	为浏览器功能测试提供 jsdom slot 测试台
loader-smoke	启动由 Loader 组合的应用并驱动 fixture 轮次以执行冒烟测试
llm-mock-server	为恢复测试提供可编脚本的 OpenAI 兼容故障服务器
llm-replay	为无密钥测试与演示回放已记录的模型流

相关文档

- 测试策略这些 harness 所服务的无密钥快照层及其适用时机。
- 运行时不变式 test-suppor `./invariant` 形式随附的包自有运行时检查。
- 包组——支持组与产品组的关系。

开发备注

维护者的工作上下文——点击展开

无。

runtime-diagnostics/ 运行时诊断：按包

---

description: "runtime-diagnostics 组地图：面向用户与维护者浏览本组的包自有运行时检查能力。"

kind: "package-group"

English | 中文

概述

runtime-diagnostics 组为 DeepSeek Harness 的 `invariants` 检查组合运行期间运行包自有检查，验证每个包的持久事件与数据关系。违规会以归因到拥有该关系的包的错误

目录

- 包
- 相关文档
- 开发备注

包

包	职责	**ctx** 键
`invariants`	运行包自有运行时检查，并按所属包报告每次失败	`ctx.invariants`

相关文档

- 变成部服务参考：选择、installer 与配套入口约定。

- [redacted] Agent Note [redacted]
- [redacted] Agent Note [redacted]
- [redacted] [redacted] `./invariant` [redacted]

[redacted]

[redacted]
[redacted]

util/: 共享工具

---

description: "共享工具家族的包映射：原子文件写入、品牌化 id、双端队列、JSON 值、harness 主目录路径、启动环境、原生命令、输出保留、时区与超时。"

kind: "package-group"

English | 中文

概述

`util/` 组为能力包提供共享的机制原语，避免重复实现。它涵盖原子写入、品牌化 id、双端队列、无损 JSON 值、UUID、Harness home 路径、启动环境、原生命令、输出保留、时区规范

目录

- 包
- 相关文档
- 开发备注

包

每个包提供一个原语；打开对应包页面了解如何使用。

包	职责
`brand/`	提供名义字符串类型及其无状态构造函数
`crypto/`	`crypto.getRandomValues` 原语生成 RFC 9562 v4 UUID
`deque/`	提供摊销常数时间的队列操作和有界空闲存储
`values/`	校验、创建快照、比较和冻结无损 JSON 兼容值
`home-paths/`	解析统一的 Harness 主目录并拼接共享的用户数据路径

包	职责
launch-environment/	冻结的启动环境，记住每个值来自哪一层
atomic-write/	原子文件替换与跨进程写锁
native-command/	直接运行宿主原生命令，绝不拼 shell 字符串
workspace-path/	提供浏览器安全的 Workspace 路径与显示辅助函数
output-retention/	限制面向模型的输出并报告精确的省略元数据
time/	校验并规范化调用方所报的 IANA 时区
timeout/	截止时间运算、信号融合与超时/取消分类

相关文档

- 根包映射 util/ 在所有包组中的位置。
- 生成配置和锁所属的库包索引。
- 添加包实现时共享原语如何落入本组。

开发备注

维护者的工作上下文——点击展开

无。