

DeepSeek Harness

[English](#) |

packages/ Cordis

agent

Cordis

[Cordis](#) dsh agent loop

dsh

Profile

dsh

profile Harness hor cordis.patch.yml web headless sdk sdk-minimal acp

Cordis patch

package.json dsh dsh.profile prc dsh.bundle patch

[dsh-base](#) web headless sdk acp profile [dsh-web-app](#) [dsh-headless](#) [dsh-sdk-app](#) SDK JSON-RP [dsh-](#)
[acp-app](#) [dsh-sdk-minimal](#) dsh-base SDK

cordis.patch.yml hor --patch overlay patch id config

profile web profile headless sdk sdk-minimal acp stdio

```
dsh --profile web --dump-config
```

patch

app-boot



dsh --profile web dsh --profile headless dsh --profile sdk dsh --profile sdk-minimal dsh --profile acp TypeScript SDK dsh sdk proc sdk-minimal launcher

Vendored CLI verify-application-entypoints Harness dsh Node

Python SDK dsh deepseek-harness-sdk-runtime-<platform>-<arch> Harness dsh --re profile sdk sdk-minimal profile Python profile patch dsh plugin ordi bin pars



Cordis

		ctx
core/session	SessionEvent	ctx.sessions
core/system-prompt	schema	ctx.systemPrompt
core/tools		ctx.tools
core/agent	Agent agent/*	ctx.agents
core/agent-loop		ctx.agentLoop
core/scope	agent	ctx
llm/llm	seam	ctx.llm
webhook/webhook	delivery Workspace Session	ctx.webhookRuntime



- session/event
- Agent** agent/* Agent inbox
- fs/* tools/* telemetry/*



```
turn/start
  claim next-step input plus one queued message
  assemble prompt sections + tool schemas
  -> agent/pre-step          reject | enter(messages, startsRequestSeries?)
    reject, or a first enter rewritten empty -> close the turn with no step
  step/start
    append entered messages as user/message
    derive model history from the log
    agent/request -> llm/stream -> assistant/chunk* -> assistant/message
    tool/call* -> tools/pre-execute -> tools/execute -> tools/post-execute -> tool/result*
  step/end
    tools owe another request, or next-step input arrived -> claim -> next step
  -> agent/turn-stopping
turn/end
```

turn/* step/* user/message assistant/* [tool/* agent/pre-step agent/request llm/stream [tools/* [waterfall
next() agent/turn-stopping [serial next()]

[] inbox [] inbox []

agent/pre-step [] startsRequestSeries [] request/header [] series [] startsSeries: true [] change enter [] {
...decision, messages } [] schema []



deriveMessages() [assistant/chunk [] UI [] fork [] transcript []

[SessionEventMap []

[] **sear** dsh-session-projection [ctx.sessionProjections [] stateOf() [] snapshot() [] host []
ctx.inject(['sessionProjections'], ...) [] host [] turnBoundary [] [] []

seam

seam Service Definition Service Provider Consumer seam

seam subagent agent Bash PTY LSP fork

Agent Team ctx.agentTeams seam subagent roster mailbox



	ctx.llm
	ctx.tools schema
	agent pre isolate realm
shell	ctx.shell ctx.subprocess spawn
	ctx.terminals dsh-tool-terminal
	ctx.commands
	ctx.jobs job_*
webhook Session	ctx.webhookRuntime
	ctx.fs fs/*
	ctx.sandbox argv
	agent/* tools/* agent/turn-stopping
	agent.inject()
UI	ctx.agents session/event
Web Client Chat	ConversationNodeDefinition + keyed renderer
	SessionEventMap
	ctx.sessionTitle
	ctx.goals agent/*
fork	ctx.sessions.fork(source, boundary?, childSessionId?)
agent	agent agent.ctx

Conversation Chat node

□□ #1

□ □□□ □□ 3 □ 2026 16:55:27

□ □□□ □□ 3 □ 2026 17:02:14