

Command Manual

Complete version-pinned syntax, options, behavior, and nested help surfaces for every public ait-native 1.1.1 CLI command.

VERSION	DOCUMENTATION
1.1.1 / 1.1.1	Revision 4
CONTENTS	CLI REFERENCE
34 reference chapters	29 top-level commands 160 non-root help surfaces

Contents

Use the linked page numbers to move through the PDF. Each chapter also links to its exact online route.

Complete 1.1.1 CLI Reference	4
ait init - Complete Reference	7
ait plan - Complete Reference	8
ait task - Complete Reference	13
ait blame - Complete Reference	16
ait doctor - Complete Reference	17
ait queue - Complete Reference	19
ait remote - Complete Reference	20
ait release - Complete Reference	22
ait repo - Complete Reference	27
ait config - Complete Reference	30
ait external - Complete Reference	33
ait status - Complete Reference	36
ait diff - Complete Reference	37
ait pull - Complete Reference	38
ait push - Complete Reference	39
ait gc - Complete Reference	40
ait stash - Complete Reference	42
ait change - Complete Reference	45
ait snapshot - Complete Reference	49
ait commit - Complete Reference	55
ait tag - Complete Reference	56
ait patchset - Complete Reference	58
ait review - Complete Reference	61
ait attest - Complete Reference	67
ait policy - Complete Reference	69

ait worktree - Complete Reference	71
ait workflow - Complete Reference	80
ait line - Complete Reference	83
ait git - Complete Reference	87
ait-agent - Complete Reference	89
ait-agent-worker - Complete Reference	99
ait-server - Complete Reference	102
ait-runner - Complete Reference	103
1.1.1 Source Authority	105

CLI · COMPLETE 1.1.1 REFERENCE

Complete 1.1.1 CLI Reference

Browse every public 1.1.1 ait command and command group with exact emitted syntax, options, defaults, conflicts, and operating boundaries.

AUDIENCE Developers, coding agents, and operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/>

Coverage

This is the complete public command reference emitted by the admitted native `ait 1.1.1` binary. It covers **29 top-level commands** and **160 non-root public command or command-group help surfaces**. The root `ait --help` surface is reproduced separately below.

Clap's automatically generated `help` aliases are navigation helpers, so they are not counted as separate product operations. Hidden and internal parser surfaces are also outside this public reference.

How to read the syntax

- `<VALUE>` is a required value supplied by the caller; do not type the angle brackets literally.
- `[OPTIONS]`, `[COMMAND]`, and other square-bracket groups are optional.
- A choice shown inside the emitted usage line expresses a parser-enforced relationship. Preserve the exact grouping when constructing a command.
- Default values, possible values, conflicts, and requirements appear exactly where the 1.1.1 binary emits them.

Top-level command index

- [`ait init`](#) - Create or reinitialize local AIT repository authority and refresh the effective repository instructions.
- [`ait plan`](#) - Record Markdown Plan lineage, inspect revisions, and synchronize exact authored Markdown in the selected scope.
- [`ait task`](#) - Start Plan-bound work, inspect and audit Task state, finish admitted work, or abandon an unfinished Task.
- [`ait blame`](#) - Attribute a path or line range to recorded Snapshot and Plan lineage, with an explicit bounded restore option.
- [`ait doctor`](#) - Run focused diagnostics for the configured memory root, runtime root, or Plan authority.
- [`ait queue`](#) - Read active work, review, remote, and worktree inventory in summary form.
- [`ait remote`](#) - Register and list remote authorities or reconstruct a local Line head from an exact remote Repository authority.
- [`ait release`](#) - Create and inspect release dossiers, build and package native artifacts, publish them, and promote admitted releases.
- [`ait repo`](#) - Inspect or administer a registered remote Repository and its CI jobs, capabilities, and runs.
- [`ait config`](#) - Inspect effective repository configuration or set an explicit supported configuration value.
- [`ait external`](#) - Inspect, materialize, pin, link, unlink, and diagnose external repository dependencies.
- [`ait status`](#) - Inspect the current Line, repository, workspace, remote, and worktree state.
- [`ait diff`](#) - Compare the current workspace with recorded state using optional bounded path and output filters.
- [`ait pull`](#) - Import a remote Line's Snapshot history, classify head relationships, and optionally materialize the selected result.
- [`ait push`](#) - Upload recorded Snapshot history and advance an exact remote Line.

- [`ait gc`](#) - Inspect, validate, and prune eligible native storage packs.
- [`ait stash`](#) - Record, inspect, apply, and remove local work-in-progress stash entries.
- [`ait change`](#) - Create, inspect, update, cancel, and otherwise manage the Change lifecycle within a Task.
- [`ait snapshot`](#) - Create and inspect immutable repository state, traverse its parent history, and replay or restore recorded content.
- [`ait commit`](#) - Create an AIT Snapshot through a Git-friendly commit-named entry point.
- [`ait tag`](#) - Create, inspect, list, and remove local AIT Tag references to recorded state.
- [`ait patchset`](#) - Publish and inspect Change candidates, select the current candidate, and manage its CI request state.
- [`ait review`](#) - Request, inspect, and record review decisions for a selected Patchset.
- [`ait attest`](#) - Create and inspect typed evidence attached to a selected Patchset.
- [`ait policy`](#) - Evaluate, inspect, and explicitly waive supported policy checks for a selected Patchset.
- [`ait worktree`](#) - Inspect, diagnose, recover, relocate, and clean AIT-owned Task worktrees and their metadata.
- [`ait workflow`](#) - Classify bounded edits, reconcile workflow state, prepare remote readiness, and drive configured closeout decisions.
- [`ait line`](#) - Inspect and manage Lines, switch or restore workspace state, merge recorded history, and clean eligible Line records.
- [`ait git`](#) - Import, export, and mirror history between Git and AIT's native repository model.

Root help

The following block is the unabridged output of `ait --help` from the captured 1.1.1 binary.

AIT native local repository and workflow tool.

Usage: ait <COMMAND>

Commands:

init	Create or reinitialize a local AIT repository.
line	Manage named Lines and their Snapshot heads. [alias: branch]
git	Import, export, or mirror history between Git and AIT.
blame	Attribute selected lines without modifying workspace files.
doctor	Inspect effective AIT configuration and host readiness without repairing it.
queue	Summarize active Tasks, Changes, worktrees, and reviews.
remote	Manage server endpoints and recover the canonical remote main head.
release	Build, inspect, and publish versioned release artifacts.
repo	Inspect and manage the configured Repository on its remote server.
config	Inspect effective configuration and change supported user settings.
external	Manage reproducible external Repository dependencies.
status	Inspect the current repository, Line, Snapshot, workspace, and actionable local status.
diff	Compare the current workspace with the current Line head.
pull	Import one remote Line and safely update its local head.
push	Upload one local Line and its Snapshot history to a remote.
gc	Inspect and safely maintain local content storage.
stash	Park and restore temporary local-only workspace Snapshots.
plan	Inspect and sync local or remote Plans.
task	Start, inspect, check, finish, or abandon Tasks locally or on a remote.
change	Create, inspect, publish, or close Task-owned Changes.
snapshot	Create and inspect immutable local Snapshots and restore selected workspace content.
commit	Create an AIT Snapshot using Git-friendly commit naming.
tag	Create, inspect, and delete local Snapshot Tags.
patchset	Publish and inspect remote Change revisions and their CI state.
review	Inspect remote Review state and record team, human Task, or AI code-review evidence.
attest	Record and inspect evidence for remote Patchsets.
policy	Check, inspect, or waive remote Patchset requirements.
worktree	Inspect, restore, recover, synchronize, rebase, and remove isolated worktrees.
workflow	Prepare Changes for review, finish ready work, and repair workflow state.
help	Print this message or the help of the given subcommand(s)

Options:

-h, --help	Print help
-V, --version	Print version

Capture authority

- Product version: 1.1.1
- Binary SHA-256 used for capture: 6412a60452d5d1ee2c6d4277c09d0804e4ad73e77da58713ec312a04a6870098
- Complete help capture SHA-256: f6c2bb51a447220ed6c6856ea56fbe87aa518b55defd68e766d440e1cefbb185
- CLI parser SHA-256: 829ccf69699c28bcca872cd2b191d256ed8aef4b4e8ada0590fb1e0aacb38215
- CLI parser source: [immutable 1.1.1 source](#)

The generated receipt stores every command path, emitted description, usage, child command list, help digest, and exact help text. Website and PDF builds validate that receipt against these pages.

CLI · COMPLETE 1.1.1 REFERENCE

ait init - Complete Reference

Create or reinitialize local AIT repository authority and refresh the effective repository instructions.

AUDIENCE Users and repository owners

<https://ait-native.dev/technical/v1.1.1/cli/reference/init/>

Operating boundary

Purpose: Create or reinitialize local AIT repository authority and refresh the effective repository instructions.

Effect class: Local repository mutation; repair behavior is explicit and bounded by the emitted options.

Coverage: 1 public help surface under `ait init`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

ait init

Exact emitted help

ait init

Binary description: Create or reinitialize a local AIT repository.

```
Create or reinitialize a local AIT repository.

Usage: ait init [OPTIONS]

Options:
  --policy-profile <POLICY_PROFILE>
      Initial policy profile: prototype, team, or release. [default: prototype]
  --repair-existing
      Complete missing structure in an existing .ait directory; invalid existing data is
      never overwritten.
  --json
      Emit the stable JSON result.
  -h, --help
      Print help
```

CLI · COMPLETE 1.1.1 REFERENCE

ait plan - Complete Reference

Record Markdown Plan lineage, inspect revisions, and synchronize exact authored Markdown in the selected scope.

AUDIENCE Coding agents and maintainers

<https://ait-native.dev/technical/v1.1.1/cli/reference/plan/>

Operating boundary

Purpose: Record Markdown Plan lineage, inspect revisions, and synchronize exact authored Markdown in the selected scope.

Effect class: Mixed read and Plan mutation surface; sync scope is explicit in the emitted options.

Coverage: 8 public help surfaces under `ait plan`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait plan
ait plan list
ait plan show
ait plan revisions
ait plan items
ait plan candidates
ait plan inspect
ait plan sync
```

Exact emitted help

ait plan

Binary description: Inspect, select, and sync Plan revision history. Without `--local` or `--remote`, `solo_local` uses local Plans and `solo_remote` uses the configured default remote. Either flag explicitly overrides that default. Remote sync updates local Plan history before publishing; no Plan command creates a Snapshot or advances a Line.

```
Inspect, select, and sync Plan revision history. Without --local or --remote, solo_local uses
local Plans and solo_remote uses the configured default remote. Either flag explicitly
overrides that default. Remote sync updates local Plan history before publishing; no Plan
command creates a Snapshot or advances a Line.
```

Usage: `ait plan <COMMAND>`

Commands:

<code>list</code>	List local Plans or Plans from a remote.
<code>show</code>	Show one Plan and its selected revision.
<code>revisions</code>	List revision history for one Plan.
<code>items</code>	List checklist items from one Plan revision.
<code>candidates</code>	Find active Plans that can supply Task candidates.
<code>inspect</code>	Inspect Task-readiness details for one Plan revision.
<code>sync</code>	Sync file-backed Markdown into Plan revision history using the workflow mode's local or remote default. <code>solo_local</code> saves local Plan changes; <code>solo_remote</code> updates local Plan history and publishes the changed heads to the configured default remote. <code>--local</code> or <code>--remote</code> overrides that default. Plan sync never creates a Snapshot or advances a Line.
<code>help</code>	Print this message or the help of the given subcommand(s)

Options:

<code>-h, --help</code>	Print help (see a summary with <code>'-h'</code>)
-------------------------	--

ait plan list

Binary description: List Plans locally, from --remote, or from the location selected by the workflow mode. Human output is bounded to active Plans unless --all is supplied; JSON always returns the complete command result and therefore rejects --all. This command is read-only.

List Plans locally, from --remote, or from the location selected by the workflow mode. Human output is bounded to active Plans unless --all is supplied; JSON always returns the complete command result and therefore rejects --all. This command is read-only.

Usage: ait plan list [OPTIONS]

Options:

--local

Use local Plans even when the workflow mode defaults to a remote

--remote <NAME>

Use Plans from the named remote even when the workflow mode defaults to local

--json

Emit the complete stable machine-readable payload

--all

Expand human output to include archived Plans; unavailable with --json

-h, --help

Print help (see a summary with '-h')

ait plan show

Binary description: Show one Plan locally, from --remote, or from the location selected by the workflow mode. Without --revision, show the current head revision. Local selectors include PR-<ordinal>, artifact:<path>, title:<title>, and published-plan:<index>; remote selectors are the identifiers returned by remote plan list. This command is read-only.

Show one Plan locally, from --remote, or from the location selected by the workflow mode. Without --revision, show the current head revision. Local selectors include PR-<ordinal>, artifact:<path>, title:<title>, and published-plan:<index>; remote selectors are the identifiers returned by remote plan list. This command is read-only.

Usage: ait plan show [OPTIONS] <PLAN>

Arguments:

<PLAN>

Plan selector; local canonical forms are PR-<ordinal>, artifact:<path>, title:<title>, or published-plan:<index>

Options:

--revision <REVISION>

Select a non-head revision; local canonical forms are plan-revision:<index>, revision-number:<number>, or published-revision:<index>

--local

Use local Plans even when the workflow mode defaults to a remote

--remote <NAME>

Use Plans from the named remote even when the workflow mode defaults to local

--json

Emit the complete stable machine-readable payload

-h, --help

Print help (see a summary with '-h')

ait plan revisions

Binary description: List one Plan's revisions newest first, using local data, --remote, or the location selected by the workflow mode. Human output is bounded unless --all is supplied; JSON already returns the complete command result and therefore rejects --all. This command is read-only.

List one Plan's revisions newest first, using local data, `--remote`, or the location selected by the workflow mode. Human output is bounded unless `--all` is supplied; JSON already returns the complete command result and therefore rejects `--all`. This command is read-only.

Usage: `ait plan revisions [OPTIONS] <PLAN>`

Arguments:

`<PLAN>`

Plan selector; local canonical forms are `PR-<ordinal>`, `artifact:<path>`, `title:<title>`, or `published-plan:<index>`

Options:

`--local`

Use local Plans even when the workflow mode defaults to a remote

`--remote <NAME>`

Use Plans from the named remote even when the workflow mode defaults to local

`--json`

Emit the complete stable machine-readable payload

`--all`

Expand human output to the complete revision history; unavailable with `--json`

`-h, --help`

Print help (see a summary with `'-h'`)

ait plan items

Binary description: List parsed checklist items from one local or remote Plan. Without `--revision`, inspect the current head. This command is read-only and does not change item or Task state.

List parsed checklist items from one local or remote Plan. Without `--revision`, inspect the current head. This command is read-only and does not change item or Task state.

Usage: `ait plan items [OPTIONS] <PLAN>`

Arguments:

`<PLAN>`

Plan selector; local canonical forms are `PR-<ordinal>`, `artifact:<path>`, `title:<title>`, or `published-plan:<index>`

Options:

`--revision <REVISION>`

Select a non-head revision; local canonical forms are `plan-revision:<index>`, `revision-number:<number>`, or `published-revision:<index>`

`--local`

Use local Plans even when the workflow mode defaults to a remote

`--remote <NAME>`

Use Plans from the named remote even when the workflow mode defaults to local

`--json`

Emit the complete stable machine-readable payload

`-h, --help`

Print help (see a summary with `'-h'`)

ait plan candidates

Binary description: Find active local or remote Plans, ordered for Task selection. By default only Plans with at least one taskable item are returned; `--all` also includes active Plans with zero taskable items. `--contains` applies case-insensitive OR matching across Plan titles, file paths and selectors, headings, and item text or refs. This command is read-only.

Find active local or remote Plans, ordered for Task selection. By default only Plans with at least one taskable item are returned; --all also includes active Plans with zero taskable items. --contains applies case-insensitive OR matching across Plan titles, file paths and selectors, headings, and item text or refs. This command is read-only.

Usage: ait plan candidates [OPTIONS]

Options:

--local

Use local Plans even when the workflow mode defaults to a remote

--remote <NAME>

Use Plans from the named remote even when the workflow mode defaults to local

--all

Also include active Plans with zero taskable items; archived Plans remain excluded

--contains <TERMS>

Case-insensitive comma-delimited OR search across titles, paths, selectors, headings, item text, and item refs

--json

Emit the complete stable machine-readable payload

-h, --help

Print help (see a summary with '-h')

ait plan inspect

Binary description: Inspect summary counts and per-item Task-readiness blockers for one local or remote Plan. Without --revision, inspect the current head. This command is read-only.

Inspect summary counts and per-item Task-readiness blockers for one local or remote Plan. Without --revision, inspect the current head. This command is read-only.

Usage: ait plan inspect [OPTIONS] <PLAN>

Arguments:

<PLAN>

Plan selector; local canonical forms are PR-<ordinal>, artifact:<path>, title:<title>, or published-plan:<index>

Options:

--revision <REVISION>

Select a non-head revision; local canonical forms are plan-revision:<index>, revision-number:<number>, or published-revision:<index>

--local

Use local Plans even when the workflow mode defaults to a remote

--remote <NAME>

Use Plans from the named remote even when the workflow mode defaults to local

--json

Emit the complete stable machine-readable payload

-h, --help

Print help (see a summary with '-h')

ait plan sync

Binary description: Sync file-backed Markdown into Plan revision history using the workflow mode's local or remote default. solo_local saves local Plan changes; solo_remote updates local Plan history and publishes the changed heads to the configured default remote. --local or --remote overrides that default. Plan sync never creates a Snapshot or advances a Line.

Sync file-backed Markdown into Plan revision history using the workflow mode's local or remote default. `solo_local` saves local Plan changes; `solo_remote` updates local Plan history and publishes the changed heads to the configured default remote. `--local` or `--remote` overrides that default. Plan sync never creates a Snapshot or advances a Line.

Usage: `ait plan sync [OPTIONS] <TARGET>`

Arguments:

`<TARGET>`

Repository-relative Markdown file or directory to sync into local Plan revision history

Options:

`--plan-ref <PLAN_REF>`

Select one exact `[plan-ref: ...]` root when one Markdown target contains multiple Plan roots

`--prune`

Archive tracked Plan artifacts missing from the complete selected target inventory

`--local`

Sync local Plan history only, even when the workflow mode defaults to a remote; never create a Snapshot or advance a Line

`--remote <NAME>`

Sync local Plan history, then publish the updated heads to the named remote even when the workflow mode defaults to local

`--rebase`

When using a remote, replay the current local Plan head after the remote rejects divergent history

`--reconcile`

Safely adopt verified divergent or mixed Plan identity before effective local reconciliation or remote publication

`--json`

Emit the complete stable machine-readable result; failed and partial-success results still exit nonzero

`-h, --help`

Print help (see a summary with `'-h'`)

CLI · COMPLETE 1.1.1 REFERENCE

ait task - Complete Reference

Start Plan-bound work, inspect and audit Task state, finish admitted work, or abandon an unfinished Task.

AUDIENCE Coding agents and maintainers

<https://ait-native.dev/technical/v1.1.1/cli/reference/task/>

Operating boundary

Purpose: Start Plan-bound work, inspect and audit Task state, finish admitted work, or abandon an unfinished Task.

Effect class: Task lifecycle and bound-workspace mutation; finish advances the configured target only after admission, while abandon performs the bounded unfinished-work closeout described by its exact help.

Coverage: 7 public help surfaces under `ait task`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait task
ait task start
ait task list
ait task show
ait task audit
ait task finish
ait task abandon
```

Exact emitted help

ait task

Binary description: Start, inspect, check, finish, or abandon Tasks locally or on a remote.

```
Start, inspect, check, finish, or abandon Tasks locally or on a remote.

Usage: ait task <COMMAND>

Commands:
  start  Start one Task and its initial Change locally or on a remote; sprint mode uses one
         specific file-backed Plan item
  list   List open Tasks or complete Task history locally or on a remote
  show   Show one local or remote Task
  audit  Check one local or remote Task against main without changing it
  finish Finish one Task or selected Change onto main. The configured workflow mode chooses
         local or remote unless --local or --remote explicitly overrides it. Dirty local work
         requires --message to create its final Snapshot; clean local work reuses the current Line
         head. Remote finish consumes an already-ready selected Patchset and rejects --message. Final
         Task cleanup removes the bound worktree and archives its accepted-head feature Line.
  abandon Permanently abandon one local or remote Task and cancel its open Changes
  help   Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

ait task start

Binary description: Start one Task and its initial Change locally or on a remote; sprint mode uses one specific file-backed Plan item

Start one Task and its initial Change locally or on a remote; sprint mode uses one specific file-backed Plan item

Usage: `ait task start --intent <INTENT> (--from <MARKDOWN_PATH#ITEM_REF> | --title <TITLE>) [--edit-root <ABSOLUTE_PATH>] [--local | --remote <REMOTE>] [--json [--full]]`

Options:

<code>--title <TITLE></code>	Unbound Task title; required and available only when sprint mode is off
<code>--intent <INTENT></code>	Required Task intent in both manual and --from modes
<code>--from <MARKDOWN_PATH#ITEM_REF></code>	Sprint-only exact file-backed Plan source; syncs, validates, binds, and derives the Task and initial Change title deterministically
<code>--edit-root <ABSOLUTE_PATH></code>	Use this safe absolute path as the Task worktree; omit it to keep automatic managed worktree allocation
<code>--local</code>	Start the Task locally even when the workflow mode defaults to a remote
<code>--remote <REMOTE></code>	Start the Task on the named remote even when the workflow mode defaults to local
<code>--json</code>	Emit the compact versioned machine-readable Task-start result
<code>--full</code>	With --json, emit the previous complete Task-start result
<code>-h, --help</code>	Print help

ait task list

Binary description: List open Tasks or complete Task history locally or on a remote

List open Tasks or complete Task history locally or on a remote

Usage: `ait task list [OPTIONS]`

Options:

<code>--local</code>	List local Tasks even when the workflow mode defaults to a remote
<code>--remote <REMOTE></code>	List Tasks from the named remote even when the workflow mode defaults to local
<code>--all</code>	Show complete Task history instead of the bounded open view
<code>--json</code>	Emit the selected Task inventory as machine-readable JSON
<code>-h, --help</code>	Print help

ait task show

Binary description: Show one local or remote Task

Show one local or remote Task

Usage: `ait task show [OPTIONS] <TASK_ID>`

Arguments:

`<TASK_ID>`

Options:

<code>--local</code>	Read the local Task even when the workflow mode defaults to a remote
<code>--remote <REMOTE></code>	Read the Task from the named remote even when the workflow mode defaults to local
<code>--json</code>	Emit the complete machine-readable Task record
<code>-h, --help</code>	Print help

ait task audit

Binary description: Check one local or remote Task against main without changing it

```

Check one local or remote Task against main without changing it

Usage: ait task audit [OPTIONS] <TASK_ID>

Arguments:
  <TASK_ID>

Options:
  --local          Check the local Task even when the workflow mode defaults to a remote
  --remote <REMOTE> Check the Task on the named remote even when the workflow mode defaults
                    to local
  --json           Emit the complete machine-readable audit and recommended action
  -h, --help       Print help

```

ait task finish

Binary description: Finish one Task or selected Change onto main. The configured workflow mode chooses local or remote unless `--local` or `--remote` explicitly overrides it. Dirty local work requires `--message` to create its final Snapshot; clean local work reuses the current Line head. Remote finish consumes an already-ready selected Patchset and rejects `--message`. Final Task cleanup removes the bound worktree and archives its accepted-head feature Line.

```

Finish one Task or selected Change onto main. The configured workflow mode chooses local or
remote unless --local or --remote explicitly overrides it. Dirty local work requires --message
to create its final Snapshot; clean local work reuses the current Line head. Remote finish
consumes an already-ready selected Patchset and rejects --message. Final Task cleanup removes
the bound worktree and archives its accepted-head feature Line.

Usage: ait task finish [OPTIONS] <TASK_OR_CHANGE_ID>

Arguments:
  <TASK_OR_CHANGE_ID> Task ID or Change ID to finish onto main. The workflow mode chooses local
                    or remote unless --local or --remote is provided.

Options:
  --message <MESSAGE> Create the final Snapshot from dirty local work before finishing;
                    clean local work reuses the current Line head. Remote finish rejects this option.
  --local          Force local draft finish even when the configured workflow mode
                    defaults to remote
  --remote <REMOTE> Force closeout through the named remote's already-ready selected
                    Patchset even when the configured workflow mode defaults to local
  --json           Emit the compact versioned machine-readable finish and closeout
                    result
  --full           With --json, emit the previous complete finish and closeout result
  -h, --help       Print help

```

ait task abandon

Binary description: Permanently abandon one local or remote Task and cancel its open Changes

```

Permanently abandon one local or remote Task and cancel its open Changes

Usage: ait task abandon [OPTIONS] <TASK_ID>

Arguments:
  <TASK_ID>

Options:
  --local          Abandon the local Task even when the workflow mode defaults to a remote
  --remote <REMOTE> Abandon the Task on the named remote even when the workflow mode
                    defaults to local
  --json           Emit the complete machine-readable terminal Task record
  -h, --help       Print help

```

CLI · COMPLETE 1.1.1 REFERENCE

ait blame - Complete Reference

Attribute a path or line range to recorded Snapshot and Plan lineage, with an explicit bounded restore option.

AUDIENCE Developers and coding agents

<https://ait-native.dev/technical/v1.1.1/cli/reference/blame/>

Operating boundary

Purpose: Attribute a path or line range to recorded Snapshot and Plan lineage, with an explicit bounded restore option.

Effect class: Read-only by default; the emitted restore option mutates the selected workspace range.

Coverage: 1 public help surface under `ait blame`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

`ait blame`

Exact emitted help

`ait blame`

Binary description: Attribute selected lines without modifying workspace files.

```
Attribute selected lines without modifying workspace files.

Usage: ait blame [OPTIONS] <PATH>

Arguments:
  <PATH>

Options:
  --line <LINE>
    Return blame for one line only.
  --start <START_LINE>
    Start line for a bounded blame range.
  --end <END_LINE>
    End line for a bounded blame range.
  --snapshot <SNAPSHOT_ID>
    Blame against one explicit immutable snapshot.
  --via-parent <VIA_PARENT_SNAPSHOT_ID>
    For an explicit merge Snapshot, follow this direct parent instead of the primary
    parent.
  --patchset <PATCHSET_ID>
    Resolve one exact published Patchset ID to its revision Snapshot before blaming.
  --remote <REMOTE_NAME>
    Remote to use when resolving --patchset.
  --plan-id <PLAN_ID>
    Select one current Markdown Plan when the same file path is tracked by multiple
    current Plans.
  --plan-ref <PLAN_REF>
    Select one current Markdown Plan by file selector or reference.
  --json

-h, --help
    Print help
```

CLI · COMPLETE 1.1.1 REFERENCE

ait doctor - Complete Reference

Run focused diagnostics for the configured memory root, runtime root, or Plan authority.

AUDIENCE Users and operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/doctor/>

Operating boundary

Purpose: Run focused diagnostics for the configured memory root, runtime root, or Plan authority.

Effect class: Read-only diagnostics; each subcommand validates one explicitly selected local authority boundary.

Coverage: 4 public help surfaces under `ait doctor`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait doctor
ait doctor memory-root
ait doctor runtime-root
ait doctor plan-authority
```

Exact emitted help

ait doctor

Binary description: Inspect effective AIT configuration and host readiness without repairing it.

```
Inspect effective AIT configuration and host readiness without repairing it.

Usage: ait doctor <COMMAND>

Commands:
  memory-root  Validate the configured memory-backed runtime root.
  runtime-root Inspect the effective server runtime-root placement.
  plan-authority Inspect the fixed Rust-native Plan storage contract.
  help         Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

ait doctor memory-root

Binary description: Validate the existing memory-backed root recorded in `task_worktree.memory_root`, plus its typed capacity and derived Task runtime location. This diagnostic never mounts, provisions, repairs, or creates the root.

```
Validate the existing memory-backed root recorded in task_worktree.memory_root, plus its typed
capacity and derived Task runtime location. This diagnostic never mounts, provisions, repairs,
or creates the root.

Usage: ait doctor memory-root [OPTIONS]

Options:
  --json
    Emit the diagnostic payload as JSON.

  -h, --help
    Print help (see a summary with '-h')
```

ait doctor runtime-root

Binary description: Inspect the server runtime root selected by AIT_RUNTIME_DATA (or legacy AIT_NATIVE_SERVER_DATA) and report whether Snapshot scans protect it.

```
Inspect the server runtime root selected by AIT_RUNTIME_DATA (or legacy AIT_NATIVE_SERVER_DATA)
and report whether Snapshot scans protect it.
```

```
Usage: ait doctor runtime-root [OPTIONS]
```

```
Options:
```

```
--json
```

```
Emit the diagnostic payload as JSON.
```

```
-h, --help
```

```
Print help (see a summary with '-h')
```

ait doctor plan-authority

Binary description: Inspect the fixed Rust-native Plan storage contract and required exports.

```
Inspect the fixed Rust-native Plan storage contract and required exports.
```

```
Usage: ait doctor plan-authority [OPTIONS]
```

```
Options:
```

```
--json
```

```
Emit the diagnostic payload as JSON.
```

```
-h, --help
```

```
Print help (see a summary with '-h')
```

CLI · COMPLETE 1.1.1 REFERENCE

ait queue - Complete Reference

Read active work, review, remote, and worktree inventory in summary form.

AUDIENCE Users and coding agents

<https://ait-native.dev/technical/v1.1.1/cli/reference/queue/>

Operating boundary

Purpose: Read active work, review, remote, and worktree inventory in summary form.

Effect class: Read-only inspection.

Coverage: 2 public help surfaces under `ait queue`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait queue
ait queue summary
```

Exact emitted help

`ait queue`

Binary description: Summarize active Tasks, Changes, worktrees, and reviews.

```
Summarize active Tasks, Changes, worktrees, and reviews.

Usage: ait queue <COMMAND>

Commands:
  summary  Show active local work and the configured remote review inbox.
  help     Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

`ait queue summary`

Binary description: Show active local work and the configured remote review inbox.

```
Show active local work and the configured remote review inbox.

Usage: ait queue summary [OPTIONS]

Options:
  --remote <REMOTE>  Read the review inbox from this configured remote instead of the
                    default remote.
  --json             Emit the complete machine-readable summary.
  -h, --help         Print help
```

CLI · COMPLETE 1.1.1 REFERENCE

ait remote - Complete Reference

Register and list remote authorities or reconstruct a local Line head from an exact remote Repository authority.

AUDIENCE Repository owners and recovery operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/remote/>

Operating boundary

Purpose: Register and list remote authorities or reconstruct a local Line head from an exact remote Repository authority.

Effect class: Registration and applied recovery mutate state; recovery without its apply flag is a preview.

Coverage: 4 public help surfaces under `ait remote`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait remote
ait remote add
ait remote list
ait remote recover-head
```

Exact emitted help

ait remote

Binary description: Manage server endpoints and recover the canonical remote main head.

```
Manage server endpoints and recover the canonical remote main head.

Usage: ait remote <COMMAND>

Commands:
  add          Register a remote using the canonical Repository directory name.
  list         List configured remotes without contacting their servers.
  recover-head Preview or atomically rebuild the exact remote main head and its full Snapshot
               ancestry.
  help         Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

ait remote add

Binary description: Register a remote using the canonical Repository directory name.

```
Register a remote using the canonical Repository directory name.

Usage: ait remote add [OPTIONS] <NAME> <URL>

Arguments:
  <NAME>  Local name used to select this remote.
  <URL>   Base URL of the AIT server.

Options:
  --default  Use this remote by default for both push and pull.
  --json     Emit the complete machine-readable JSON result.
  -h, --help Print help
```

ait remote list

Binary description: List configured remotes without contacting their servers.

```
List configured remotes without contacting their servers.

Usage: ait remote list [OPTIONS]

Options:
  --json  Emit the complete machine-readable JSON result.
  -h, --help  Print help
```

ait remote recover-head

Binary description: Preview or atomically rebuild the exact remote main head and its full Snapshot ancestry.

```
Preview or atomically rebuild the exact remote main head and its full Snapshot ancestry.

Usage: ait remote recover-head [OPTIONS]

Options:
  --remote <NAME>  Read from this remote; otherwise use the configured default remote.
  --jobs <JOBS>    Use this many parallel pack downloads (1 through 64). [default: 8]
  --apply          Activate the reconstructed generation; omission performs a read-only
  preview.
  --json           Emit the complete machine-readable JSON result.
  -h, --help       Print help
```

CLI · COMPLETE 1.1.1 REFERENCE

ait release - Complete Reference

Create and inspect release dossiers, build and package native artifacts, publish them, and promote admitted releases.

AUDIENCE Release engineers

<https://ait-native.dev/technical/v1.1.1/cli/reference/release/>

Operating boundary

Purpose: Create and inspect release dossiers, build and package native artifacts, publish them, and promote admitted releases.

Effect class: Release-authority and publication surface; build, publish, and promote operations can have external effects.

Coverage: 12 public help surfaces under `ait release`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait release
ait release candidate
ait release candidate create
ait release check
ait release build
ait release native-source
ait release native-bundle
ait release package
ait release formula
ait release show
ait release publish
ait release promote
```

Exact emitted help

ait release

Binary description: Build, inspect, and publish versioned release artifacts.

```
Build, inspect, and publish versioned release artifacts.

Usage: ait release <COMMAND>

Commands:
  candidate
  check
  build
  native-source
  native-bundle
  package
  formula
  show
  publish
  promote
  help          Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

ait release candidate

Reference description: Group the operations that create release-candidate authority records.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```
Usage: ait release candidate <COMMAND>

Commands:
  create
  help    Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

ait release candidate create

Reference description: Create a release candidate for an exact version, Line, profile, and release channel.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```
Usage: ait release candidate create [OPTIONS] --version <VERSION>

Options:
  --version <VERSION>
  --line <LINE_NAME>    [default: main]
  --profile <PROFILE>
  --channel <CHANNEL>   [possible values: rc, stable]
  --json
  -h, --help            Print help
```

ait release check

Reference description: Validate a release dossier, its component receipts and artifact bytes, and the stated test evidence.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```
Usage: ait release check [OPTIONS] <RELEASE_ID>

Arguments:
  <RELEASE_ID>

Options:
  --receipts <DIR>
    Verify family component receipts and artifact bytes below DIR
  --tests-command <TESTS_COMMAND>
  --skip-tests-reason <SKIP_TESTS_REASON>
  --json
  -h, --help
    Print help
```

ait release build

Reference description: Freeze admitted family artifacts for a release from component receipts or an exact native target matrix.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```
Usage: ait release build [OPTIONS] <RELEASE_ID>

Arguments:
  <RELEASE_ID>

Options:
  --receipts <DIR>          Freeze family artifacts verified from component receipts below
                             DIR
  --native-matrix-dir <DIR>  Read target artifacts from DIR/<rust-target-triple>/release
                             using ait-native-source.json descriptors.
  --native-command-dir <DIR> Read current-host release command artifacts from this explicit
                             directory.
  --json
  -h, --help                Print help
```

ait release native-source

Reference description: Record the exact source, runner, image, Rust toolchain, and compiler inputs for one native release target.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```
Usage: ait release native-source [OPTIONS] --target <TARGET> --source-dir <DIR> --runner
<RUNNER> --runner-image <RUNNER_IMAGE> --rust-toolchain <RUST_TOOLCHAIN> --rustc-path <PATH>
<RELEASE_ID>

Arguments:
  <RELEASE_ID>

Options:
  --target <TARGET>
  --source-dir <DIR>
  --runner <RUNNER>
  --runner-image <RUNNER_IMAGE>
  --rust-toolchain <RUST_TOOLCHAIN>
  --rustc-path <PATH>
  --json
  -h, --help                Print help
```

ait release native-bundle

Reference description: Build native release bundles only from target directories carrying the required source descriptors.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```
Usage: ait release native-bundle [OPTIONS] --native-matrix-dir <DIR> <RELEASE_ID>

Arguments:
  <RELEASE_ID>

Options:
  --native-matrix-dir <DIR>  Build native bundles only from DIR/<rust-target-triple>/release
                             using ait-native-source.json descriptors.
  --json
  -h, --help                Print help
```

ait release package

Reference description: Create channel-specific package output for an admitted release.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```
Usage: ait release package [OPTIONS] --channel <CHANNEL> <RELEASE_ID>

Arguments:
  <RELEASE_ID>

Options:
  --channel <CHANNEL>  Assemble one declared channel from an immutable frozen family build
                        [possible values: homebrew, apt, winget, pypi, npm]
  --json
  -h, --help           Print help
```

ait release formula

Reference description: Generate a named package formula from an admitted release.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```
Usage: ait release formula [OPTIONS] --name <NAME> <RELEASE_ID>

Arguments:
  <RELEASE_ID>

Options:
  --name <NAME>

  --python-formula <PYTHON_FORMULA>
                        Homebrew Python dependency formula written into the generated formula. [default:
                        python@3]
  --json
  -h, --help           Print help
```

ait release show

Reference description: Show the release dossier and its recorded admission state.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```
Usage: ait release show [OPTIONS] <RELEASE_ID>

Arguments:
  <RELEASE_ID>

Options:
  --remote <REMOTE>
  --json
  -h, --help           Print help
```

ait release publish

Reference description: Publish the selected admitted release artifacts through the configured publication boundary.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```
Usage: ait release publish [OPTIONS] <RELEASE_ID>

Arguments:
  <RELEASE_ID>

Options:
  --remote <REMOTE>
  --json
  -h, --help           Print help
```

ait release promote

Reference description: Promote an admitted release to the selected release channel.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```
Usage: ait release promote [OPTIONS] --channel <CHANNEL> <RELEASE_ID>
```

```
Arguments:  
  <RELEASE_ID>
```

```
Options:  
  --channel <CHANNEL> [possible values: rc, stable]  
  --json  
  -h, --help          Print help
```

CLI · COMPLETE 1.1.1 REFERENCE

ait repo - Complete Reference

Inspect or administer a registered remote Repository and its CI jobs, capabilities, and runs.

AUDIENCE Repository owners and operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/repo/>

Operating boundary

Purpose: Inspect or administer a registered remote Repository and its CI jobs, capabilities, and runs.

Effect class: Mixed remote read and administration surface; retirement, restoration, and CI scheduling mutate remote authority.

Coverage: 6 public help surfaces under `ait repo`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait repo
ait repo show
ait repo retire
ait repo restore
ait repo jobs
ait repo ci-capabilities
```

Exact emitted help

ait repo

Binary description: Inspect and manage the configured Repository on its remote server.

```
Inspect and manage the configured Repository on its remote server.

Usage: ait repo <COMMAND>

Commands:
  show          Read Repository identity, lifecycle, storage validation, and sync state.
  retire        Drain, archive, verify, and purge a remote Repository, or abort that
                 retirement.
  restore       Restore a complete local retirement archive as a new remote Repository.
  jobs          Read one Worker Job or a bounded, optionally filtered Job inventory.
  ci-capabilities Inspect server, native runner, and remote-sync prerequisites for Patchset CI.
  help         Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

ait repo show

Binary description: Read Repository identity, lifecycle, storage validation, and sync state.

```
Read Repository identity, lifecycle, storage validation, and sync state.

Usage: ait repo show [OPTIONS]

Options:
  --remote <NAME> Use this configured remote; omission uses the Repository default remote.
  --json          Emit the complete machine-readable JSON result.
  -h, --help      Print help
```

ait repo retire

Binary description: Without `--abort`, stop new work for the remote Repository, download and verify its complete archive, then delete its server data. An unrelated complete local archive blocks retirement and must be handled with `ait repo restore --remote <NAME>`; there is no archive replacement option. With `--abort`, reactivate the Repository and preserve any complete local archive.

Without `--abort`, stop new work for the remote Repository, download and verify its complete archive, then delete its server data. An unrelated complete local archive blocks retirement and must be handled with `ait repo restore --remote <NAME>`; there is no archive replacement option. With `--abort`, reactivate the Repository and preserve any complete local archive.

Usage: `ait repo retire [OPTIONS]`

Options:

`--remote <NAME>`

Mutate this configured remote; omission uses the Repository default remote.

`--abort`

Abort an in-progress retirement and reactivate the Repository while preserving any complete local archive.

`--json`

Emit the complete machine-readable JSON result.

`-h, --help`

Print help (see a summary with `'-h'`)

ait repo restore

Binary description: Verify the complete `.ait/remote/<remote>/` retirement archive, create a new remote Repository index, upload and save its data, then update the locally configured Repository index. The archive determines the restored identity; there is no name, index, or force override.

Verify the complete `.ait/remote/<remote>/` retirement archive, create a new remote Repository index, upload and save its data, then update the locally configured Repository index. The archive determines the restored identity; there is no name, index, or force override.

Usage: `ait repo restore [OPTIONS]`

Options:

`--remote <NAME>`

Use this configured remote; omission uses the Repository default remote.

`--json`

Emit the complete machine-readable JSON result.

`-h, --help`

Print help (see a summary with `'-h'`)

ait repo jobs

Binary description: Read one Worker Job or a bounded, optionally filtered Job inventory.

Read one Worker Job or a bounded, optionally filtered Job inventory.

Usage: `ait repo jobs [OPTIONS]`

Options:

`--remote <NAME>`

Read from this configured remote; omission uses the Repository default remote.

`--worker-job-index <INDEX>`

Read one specific Worker Job from the current Repository;

cannot be combined with list filters.

`--state <STATE>`

Return only queued, running, succeeded, or failed Jobs.

[possible values: queued, running, succeeded, failed]

`--limit <COUNT>`

Return at most this many Jobs in list mode (1 through 1000;

default 50). [default: 50]

`--json`

Emit the complete machine-readable JSON result.

`-h, --help`

Print help

ait repo ci-capabilities

Binary description: Inspect server, native runner, and remote-sync prerequisites for Patchset CI.

Inspect server, native runner, and remote-sync prerequisites for Patchset CI.

Usage: ait repo ci-capabilities [OPTIONS]

Options:

--remote <NAME>	Use this configured remote; omission uses the Repository default remote.
--json	Emit the complete machine-readable JSON result.
-h, --help	Print help

CLI · COMPLETE 1.1.1 REFERENCE

ait config - Complete Reference

Inspect effective repository configuration or set an explicit supported configuration value.

AUDIENCE Users and operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/config/>

Operating boundary

Purpose: Inspect effective repository configuration or set an explicit supported configuration value.

Effect class: Show is read-only; set changes repository configuration and can refresh generated instructions.

Coverage: 4 public help surfaces under `ait config`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait config
ait config show
ait config set
ait config unset
```

Exact emitted help

ait config

Binary description: Inspect effective configuration and change supported user settings.

```
Inspect effective configuration and change supported user settings.

Usage: ait config <COMMAND>

Commands:
  show  Inspect effective repository configuration without modifying it.
  set    Set supported user-owned repository defaults.
  unset  Remove one supported user setting.
  help   Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

ait config show

Binary description: Read repository configuration plus the applicable worktree settings and report the effective values. Human output is compact; `--json` returns the complete machine-readable result.

```
Read repository configuration plus the applicable worktree settings and report the effective
values. Human output is compact; --json returns the complete machine-readable result.

Usage: ait config show [OPTIONS]

Options:
  --json
    Emit the complete machine-readable effective configuration.

  -h, --help
    Print help (see a summary with '-h')
```

ait config set

Binary description: Set one or more supported user-owned defaults, then refresh effective configuration and generated workflow guidance. Internal Repository indexes, derived local/remote settings, and Plan bindings cannot be set here. Use `ait config unset <KEY>` to remove an optional override.

Set one or more supported user-owned defaults, then refresh effective configuration and generated workflow guidance. Internal Repository indexes, derived local/remote settings, and Plan bindings cannot be set here. Use ``ait config unset <KEY>`` to remove an optional override.

Usage: `ait config set [OPTIONS]`

Options:

```
--workflow-mode <WORKFLOW_MODE>
    Set the complete workflow preset. This chooses where Workflow, Task, and Change
    commands run and defaults sprint mode to on unless --sprint is supplied.

    [possible values: solo_local, solo_remote, team_remote]

--sprint <SPRINT>
    Set sprint-style Plan/Task binding. On requires exact Plan item refs; off uses unbound
    Tasks.

    [possible values: on, off]

--default-author-mode <DEFAULT_AUTHOR_MODE>
    Set the default provenance author mode.

    [possible values: human_only, human_with_ai_assist, ai_with_human_review,
    ai_only_experimental]

--default-model <DEFAULT_MODEL>
    Set a non-empty default provenance model name.

--task-review <TASK_REVIEW>
    Set Task outcome review policy. Required waits for review; automatic records
    task_approve using configured user_name.

    [possible values: required, automatic]

--task-worktree-alias-root <TASK_WORKTREE_ALIAS_ROOT>
    Set a non-empty managed alias root for task worktrees. Relative paths resolve from the
    Repository root.

--task-worktree-main-seed-ram-max-bytes <TASK_WORKTREE_MAIN_SEED_RAM_MAX_BYTES>
    Set a non-negative Repository-local RAM budget in bytes for main-seed-backed task
    worktree bootstrap.

--id-namespace-prefix <ID_NAMESPACE_PREFIX>
    Set a non-empty ASCII-alphanumeric namespace prefix before workflow type codes such as
    T/C/P/PL/PR.

--user-name <USER_NAME>
    Set the non-empty human identity used by explicit and automatic Task review.

--user-email <USER_EMAIL>
    Set a non-empty local actor email for non-Task-review identity surfaces.

--json
    Emit the complete machine-readable effective configuration after the update.

-h, --help
    Print help (see a summary with '-h')
```

ait config unset

Binary description: Delete exactly one optional user setting from repository configuration, then refresh effective configuration and generated workflow guidance. Fallbacks are: `default-author-mode` -> `ai_with_human_review`; `default-model` -> `unset`; `task-review` -> `automatic`; `task-worktree-alias-root` -> `.ait-worktree-links`; `task-worktree-main-seed-ram-max-bytes` -> no configured budget; `id-namespace-prefix` -> `empty`; `user-name` and `user-email` -> `unset` while actor detection remains available. Internal Repository indexes, workflow-mode, sprint, derived local/remote settings, and Plan bindings cannot be unset.

Delete exactly one optional user setting from repository configuration, then refresh effective configuration and generated workflow guidance. Fallbacks are: default-author-mode -> ai_with_human_review; default-model -> unset; task-review -> automatic; task-worktree-alias-root -> .ait-worktree-links; task-worktree-main-seed-ram-max-bytes -> no configured budget; id-namespace-prefix -> empty; user-name and user-email -> unset while actor detection remains available. Internal Repository indexes, workflow-mode, sprint, derived local/remote settings, and Plan bindings cannot be unset.

Usage: ait config unset [OPTIONS] <KEY>

Arguments:

<KEY>

Supported optional user setting to remove.

[possible values: default-author-mode, default-model, task-review, task-worktree-alias-root, task-worktree-main-seed-ram-max-bytes, id-namespace-prefix, user-name, user-email]

Options:

--json

Emit the complete machine-readable effective configuration after removal.

-h, --help

Print help (see a summary with '-h')

CLI · COMPLETE 1.1.1 REFERENCE

ait external - Complete Reference

Inspect, materialize, pin, link, unlink, and diagnose external repository dependencies.

AUDIENCE Developers and dependency operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/external/>

Operating boundary

Purpose: Inspect, materialize, pin, link, unlink, and diagnose external repository dependencies.

Effect class: Status and doctor inspect; update, link, and unlink change dependency or workspace materialization state.

Coverage: 6 public help surfaces under `ait external`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait external
ait external update
ait external status
ait external doctor
ait external link
ait external unlink
```

Exact emitted help

ait external

Binary description: Inspect, diagnose, resolve, pin, restore, and locally link external Repository dependencies declared by `ait-external.toml`. Resolution writes an exact `ait-external.lock`; generated content is placed at each repository-relative destination declared in the manifest.

```
Inspect, diagnose, resolve, pin, restore, and locally link external Repository dependencies
declared by ait-external.toml. Resolution writes an exact ait-external.lock; generated content
is placed at each repository-relative destination declared in the manifest.
```

```
Usage: ait external <COMMAND>
```

Commands:

```
update  Resolve and restore declared external Repositories.
status  Inspect external pins, links, lock drift, and restored content.
doctor  Evaluate external dependency release readiness without repairing it.
link     Use another local checkout for one declared direct external.
unlink   Remove one local external override and restore locked content.
help     Print this message or the help of the given subcommand(s)
```

Options:

```
-h, --help
        Print help (see a summary with '-h')
```

ait external update

Binary description: Resolve and restore declared external Repositories. With no NAME, preserve the manifest's exact pins while updating the complete dependency lock. NAME requires exactly one of `--to` or `--latest`. `--locked` is a separate target-free mode that reads an existing drift-free lockfile without changing the manifest or lockfile.

Resolve and restore declared external Repositories. With no NAME, preserve the manifest's exact pins while updating the complete dependency lock. NAME requires exactly one of --to or --latest. --locked is a separate target-free mode that reads an existing drift-free lockfile without changing the manifest or lockfile.

```
Usage: ait external update [--validate] [--no-recursive] [--json]
      ait external update <NAME> (--to <SNAPSHOT> | --latest) [--validate] [--no-recursive]
      [--json]
      ait external update --locked [--validate] [--no-recursive] [--json]
```

Arguments:

[NAME]
Unique direct external name; requires --to or --latest.

Options:

--to <SNAPSHOT>
Pin NAME to this exact immutable Snapshot before resolving and restoring its files.

--latest
Resolve NAME's declared remote and line head, then persist that exact Snapshot pin.

--locked
Restore external files from the existing drift-free ait-external.lock without resolving or changing pins; active local links are rejected.

--validate
Prepare the selected files and run destination toolchain checks first; install them only when validation has no errors.

--no-recursive
Restore direct external dependencies only; the resolved lockfile still records the complete dependency graph.

--json
Emit the complete machine-readable update report.

-h, --help
Print help (see a summary with '-h')

ait external status

Binary description: Read the manifest, lockfile, local links, destination paths, and generated content without repairing or changing them. The report explains whether content is present, missing, linked, dirty, or outdated; lock drift is reported separately.

Read the manifest, lockfile, local links, destination paths, and generated content without repairing or changing them. The report explains whether content is present, missing, linked, dirty, or outdated; lock drift is reported separately.

```
Usage: ait external status [OPTIONS]
```

Options:

--json
Emit the complete machine-readable status report.

-h, --help
Print help (see a summary with '-h')

ait external doctor

Binary description: Check the manifest, lockfile, restored content, local links, destination bindings, licenses, and applicable current-source readiness. The report is always emitted. By default findings are diagnostic only; --fail-on-blocking returns exit code 2 when release_ready is false.

Check the manifest, lockfile, restored content, local links, destination bindings, licenses, and applicable current-source readiness. The report is always emitted. By default findings are diagnostic only; `--fail-on-blocking` returns exit code 2 when `release_ready` is false.

Usage: `ait external doctor [OPTIONS]`

Options:

`--fail-on-blocking`
Return exit code 2 after emitting the report when release-blocking findings exist.

`--json`
Emit the complete machine-readable readiness report.

`-h, --help`
Print help (see a summary with `'-h'`)

ait external link

Binary description: Record a local development override in `ait-external.links.toml`. NAME must match exactly one direct `ait-external.toml` declaration. PATH must be an existing directory outside the current Repository; regular updates preserve the override, while locked restores and release checks reject active links.

Record a local development override in `ait-external.links.toml`. NAME must match exactly one direct `ait-external.toml` declaration. PATH must be an existing directory outside the current Repository; regular updates preserve the override, while locked restores and release checks reject active links.

Usage: `ait external link [OPTIONS] <NAME> <PATH>`

Arguments:

`<NAME>`
Unique direct external name declared by `ait-external.toml`.

`<PATH>`
Existing local checkout directory; relative paths resolve from the Repository root.

Options:

`--json`
Emit the complete machine-readable link result.

`-h, --help`
Print help (see a summary with `'-h'`)

ait external unlink

Binary description: Remove NAME from `ait-external.links.toml`. When `ait-external.lock` exists, restore the matching direct external and its recursive subtree before committing removal of the override. A failed restore leaves the logical link active.

Remove NAME from `ait-external.links.toml`. When `ait-external.lock` exists, restore the matching direct external and its recursive subtree before committing removal of the override. A failed restore leaves the logical link active.

Usage: `ait external unlink [OPTIONS] <NAME>`

Arguments:

`<NAME>`
Local external override name to remove.

Options:

`--json`
Emit the complete machine-readable unlink and restoration result.

`-h, --help`
Print help (see a summary with `'-h'`)

CLI · COMPLETE 1.1.1 REFERENCE

ait status - Complete Reference

Inspect the current Line, repository, workspace, remote, and worktree state.

AUDIENCE Users and coding agents

<https://ait-native.dev/technical/v1.1.1/cli/reference/status/>

Operating boundary

Purpose: Inspect the current Line, repository, workspace, remote, and worktree state.

Effect class: Read-only inspection.

Coverage: 1 public help surface under `ait status`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait status
```

Exact emitted help

`ait status`

Binary description: Inspect the current repository, Line, Snapshot, workspace, bounded maintenance findings, and actionable repair state without changing repository data or runtime state. Use `--json` for compact versioned machine-readable output; add `--full` for the complete output.

```
Inspect the current repository, Line, Snapshot, workspace, bounded maintenance findings, and
actionable repair state without changing repository data or runtime state. Use --json for
compact versioned machine-readable output; add --full for the complete output.
```

```
Usage: ait status [OPTIONS]
```

```
Options:
```

```
--json
    Emit compact versioned machine-readable status.
```

```
--full
    With --json, emit the complete status result.
```

```
-h, --help
    Print help (see a summary with '-h')
```

CLI · COMPLETE 1.1.1 REFERENCE

ait diff - Complete Reference

Compare the current workspace with recorded state using optional bounded path and output filters.

AUDIENCE Users and coding agents

<https://ait-native.dev/technical/v1.1.1/cli/reference/diff/>

Operating boundary

Purpose: Compare the current workspace with recorded state using optional bounded path and output filters.

Effect class: Read-only inspection.

Coverage: 1 public help surface under `ait diff`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

ait diff

Exact emitted help

ait diff

Binary description: Compare the current workspace with the current Line head without changing repository data or workspace files. With no output option, emit a unified text diff. Use `--json` for stable machine-readable output, `--stat` for per-file text statistics, or `--name-only` for the ordered changed-path list. Positional `PATH` values select specific workspace-relative files or directory subtrees using lexical matching, not globs.

```
Compare the current workspace with the current Line head without changing repository data or
workspace files. With no output option, emit a unified text diff. Use --json for stable
machine-readable output, --stat for per-file text statistics, or --name-only for the ordered
changed-path list. Positional PATH values select specific workspace-relative files or
directory subtrees using lexical matching, not globs.
```

```
Usage: ait diff [OPTIONS] [PATH]...
```

```
Arguments:
```

```
  [PATH]...
```

```
    Limit the comparison to an exact workspace-relative file or directory subtree. Repeat
    PATH to combine lexical filters; glob syntax is not supported.
```

```
Options:
```

```
  --json
```

```
    Emit the stable machine-readable workspace diff.
```

```
  --stat
```

```
    Emit per-file text statistics.
```

```
  --name-only
```

```
    Emit only the ordered changed-path list.
```

```
-h, --help
```

```
    Print help (see a summary with '-h')
```

CLI · COMPLETE 1.1.1 REFERENCE

ait pull - Complete Reference

Import a remote Line's Snapshot history, classify head relationships, and optionally materialize the selected result.

AUDIENCE Developers and coding agents

<https://ait-native.dev/technical/v1.1.1/cli/reference/pull/>

Operating boundary

Purpose: Import a remote Line's Snapshot history, classify head relationships, and optionally materialize the selected result.

Effect class: Imports local authority; merge and restore options can also change Line or workspace state.

Coverage: 1 public help surface under `ait pull`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait pull
```

Exact emitted help

`ait pull`

Binary description: Import the selected remote Line's reachable Snapshot chain and safely create or fast-forward its local Line. The configured default remote and current local Line are used when omitted. Workspace files remain unchanged unless `--restore` is supplied.

```
Import the selected remote Line's reachable Snapshot chain and safely create or fast-forward its
local Line. The configured default remote and current local Line are used when omitted.
Workspace files remain unchanged unless --restore is supplied.
```

```
Usage: ait pull [OPTIONS]
```

```
Options:
```

```
--remote <REMOTE>
```

```
    Use this configured remote; defaults to the repository's default remote.
```

```
--line <LINE>
```

```
    Import this remote Line and safely create or fast-forward its local Line; defaults to
    the current local Line.
```

```
--merge
```

```
    Merge a divergent imported remote head into the current local Line and restore the
    resulting files; requires --restore and a clean workspace.
```

```
--restore
```

```
    Restore the pulled Line's files into the workspace and select that Line; rejected when
    the local Line is ahead of the remote.
```

```
--force
```

```
    Allow --restore to overwrite local workspace changes; requires --restore and cannot be
    used with --merge.
```

```
--json
```

```
    Emit the pull result as machine-readable JSON.
```

```
-h, --help
```

```
    Print help (see a summary with '-h')
```

CLI · COMPLETE 1.1.1 REFERENCE

ait push - Complete Reference

Upload recorded Snapshot history and advance an exact remote Line.

AUDIENCE Developers and recovery operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/push/>

Operating boundary

Purpose: Upload recorded Snapshot history and advance an exact remote Line.

Effect class: Remote mutation; it transfers recorded state and does not capture unrecorded workspace content.

Coverage: 1 public help surface under `ait push`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait push
```

Exact emitted help

`ait push`

Binary description: Upload an already-recorded local Line head and its required Snapshot history, then update the remote Line. Push does not capture workspace changes, create a Snapshot, publish a Patchset, or finish a Task.

```
Upload an already-recorded local Line head and its required Snapshot history, then update the
remote Line. Push does not capture workspace changes, create a Snapshot, publish a Patchset,
or finish a Task.
```

```
Usage: ait push [OPTIONS]
```

```
Options:
```

```
--remote <REMOTE>
    Upload to this configured remote instead of the repository default.
```

```
--line <LINE>
    Upload this local Line instead of the current Line.
```

```
--json
    Emit the complete machine-readable push result.
```

```
-h, --help
    Print help (see a summary with '-h')
```

CLI · COMPLETE 1.1.1 REFERENCE

ait gc - Complete Reference

Inspect, validate, and prune eligible native storage packs.

AUDIENCE Repository owners and storage operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/gc/>

Operating boundary

Purpose: Inspect, validate, and prune eligible native storage packs.

Effect class: Stats and validate inspect; prune removes only data admitted by its exact eligibility rules.

Coverage: 4 public help surfaces under `ait gc`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait gc
ait gc stats
ait gc validate
ait gc prune
```

Exact emitted help

`ait gc`

Binary description: Inspect bounded local content statistics, run exact read-only validation, or preview safe orphan object-pack cleanup. Pruning is read-only unless `gc prune` is given `--apply`; it never removes tree packs, individual unreachable blobs, Snapshots, Lines, Tasks, workspace files, or remote content.

```
Inspect bounded local content statistics, run exact read-only validation, or preview safe orphan
object-pack cleanup. Pruning is read-only unless gc prune is given --apply; it never removes
tree packs, individual unreachable blobs, Snapshots, Lines, Tasks, workspace files, or remote
content.
```

```
Usage: ait gc <COMMAND>
```

```
Commands:
```

```
stats      Show a bounded local content-storage summary.
validate   Run exact read-only local content validation.
prune      Preview safe orphan object-pack cleanup; mutate only with --apply.
help       Print this message or the help of the given subcommand(s)
```

```
Options:
```

```
-h, --help      Print help (see a summary with '-h')
```

`ait gc stats`

Binary description: Show bounded local blob, tree, and pack counts and storage metrics without exact retained-tree traversal or full inventory rows. Use `gc validate` for exact retained-Snapshot reachability.

```
Show bounded local blob, tree, and pack counts and storage metrics without exact retained-tree traversal or full inventory rows. Use gc validate for exact retained-Snapshot reachability.
```

```
Usage: ait gc stats [OPTIONS]
```

```
Options:
```

```
--json  
    Emit the bounded summary as stable JSON.
```

```
-h, --help  
    Print help (see a summary with '-h')
```

ait gc validate

Binary description: Run exact read-only validation by traversing retained Snapshot trees and blobs and checking local pack metadata without modifying repository or workspace state. The result is emitted before the command returns nonzero when `needs_attention` is true.

```
Run exact read-only validation by traversing retained Snapshot trees and blobs and checking local pack metadata without modifying repository or workspace state. The result is emitted before the command returns nonzero when needs_attention is true.
```

```
Usage: ait gc validate [OPTIONS]
```

```
Options:
```

```
--json  
    Emit the exact validation result as stable JSON.
```

```
-h, --help  
    Print help (see a summary with '-h')
```

ait gc prune

Binary description: Preview the exact fully orphaned object-pack cleanup plan by default. `--apply` recomputes, verifies, and revalidates that plan before changing local catalog metadata and deleting unreferenced object-pack archives. This command never prunes tree packs, individual unreachable blobs, Snapshots, Lines, Tasks, workspace files, or remote content.

```
Preview the exact fully orphaned object-pack cleanup plan by default. --apply recomputes, verifies, and revalidates that plan before changing local catalog metadata and deleting unreferenced object-pack archives. This command never prunes tree packs, individual unreachable blobs, Snapshots, Lines, Tasks, workspace files, or remote content.
```

```
Usage: ait gc prune [OPTIONS]
```

```
Options:
```

```
--apply  
    Apply the recomputed and revalidated safe plan; without this option the command is read-only.
```

```
--json  
    Emit the preview or apply result as stable JSON.
```

```
-h, --help  
    Print help (see a summary with '-h')
```

CLI · COMPLETE 1.1.1 REFERENCE

ait stash - Complete Reference

Record, inspect, apply, and remove local work-in-progress stash entries.

AUDIENCE Developers and coding agents

<https://ait-native.dev/technical/v1.1.1/cli/reference/stash/>

Operating boundary

Purpose: Record, inspect, apply, and remove local work-in-progress stash entries.

Effect class: Local stash and workspace mutation; list and show are read-only.

Coverage: 7 public help surfaces under `ait stash`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait stash
ait stash save
ait stash list
ait stash show
ait stash apply
ait stash pop
ait stash drop
```

Exact emitted help

ait stash

Binary description: Park, inspect, restore, and drop temporary local-only workspace Snapshots without advancing Line heads. A stash can be restored only while its source Line is current.

Park, inspect, restore, and drop temporary local-only workspace Snapshots without advancing Line heads. A stash can be restored only while its source Line is current.

Usage: `ait stash <COMMAND>`

Commands:

```
save  Save modified workspace content as a temporary local-only stash.
list  List active local-only stash metadata without changing the workspace.
show  Inspect metadata for one active local-only stash.
apply Restore a same-Line stash and retain its stash record.
pop   Restore a same-Line stash and then drop its stash record.
drop  Drop an active stash record without changing workspace content.
help  Print this message or the help of the given subcommand(s)
```

Options:

```
-h, --help
      Print help (see a summary with '-h')
```

ait stash save

Binary description: Save modified managed-workspace content as a temporary local-only stash Snapshot without advancing the current Line head. By default, restore the current Line head into the workspace after saving; `--keep-workspace` leaves the saved files in place.

Save modified managed-workspace content as a temporary local-only stash Snapshot without advancing the current Line head. By default, restore the current Line head into the workspace after saving; `--keep-workspace` leaves the saved files in place.

Usage: `ait stash save [OPTIONS]`

Options:

`--message <MESSAGE>`
Record an optional human-readable message with the stash.

`--keep-workspace`
Leave the saved files in the workspace; the Line head stays unchanged, so the workspace remains dirty relative to it.

`--json`
Emit the complete machine-readable JSON result.

`-h, --help`
Print help (see a summary with `'-h'`)

ait stash list

Binary description: List active local-only stash metadata without changing the workspace.

List active local-only stash metadata without changing the workspace.

Usage: `ait stash list [OPTIONS]`

Options:

`--json` Emit the complete machine-readable JSON result.

`-h, --help` Print help

ait stash show

Binary description: Inspect metadata for one active local-only stash without changing the workspace. This command reports the stash record and Snapshot summary; it does not display a content diff.

Inspect metadata for one active local-only stash without changing the workspace. This command reports the stash record and Snapshot summary; it does not display a content diff.

Usage: `ait stash show [OPTIONS] <STASH_ID>`

Arguments:

`<STASH_ID>`
Exact active stash ID.

Options:

`--json`
Emit the complete machine-readable JSON result.

`-h, --help`
Print help (see a summary with `'-h'`)

ait stash apply

Binary description: Replace the entire managed workspace with an active stash Snapshot and retain its stash record, without moving the current Line head. The current Line must be the stash's source Line. This restores the complete workspace rather than applying a patch or three-way merge.

Replace the entire managed workspace with an active stash Snapshot and retain its stash record, without moving the current Line head. The current Line must be the stash's source Line. This restores the complete workspace rather than applying a patch or three-way merge.

Usage: `ait stash apply [OPTIONS] <STASH_ID>`

Arguments:

`<STASH_ID>`

Exact active stash ID created on the current Line.

Options:

`--force`

Overwrite unsaved managed-workspace changes; this does not permit restoring a stash from another Line.

`--json`

Emit the complete machine-readable JSON result.

`-h, --help`

Print help (see a summary with `'-h'`)

ait stash pop

Binary description: Replace the entire managed workspace with an active stash Snapshot and drop its stash record only after a successful restore, without moving the current Line head. The current Line must be the stash's source Line. This restores the complete workspace rather than applying a patch or three-way merge.

Replace the entire managed workspace with an active stash Snapshot and drop its stash record only after a successful restore, without moving the current Line head. The current Line must be the stash's source Line. This restores the complete workspace rather than applying a patch or three-way merge.

Usage: `ait stash pop [OPTIONS] <STASH_ID>`

Arguments:

`<STASH_ID>`

Exact active stash ID created on the current Line.

Options:

`--force`

Overwrite unsaved managed-workspace changes; this does not permit restoring a stash from another Line.

`--json`

Emit the complete machine-readable JSON result.

`-h, --help`

Print help (see a summary with `'-h'`)

ait stash drop

Binary description: Drop an active stash record without changing workspace content.

Drop an active stash record without changing workspace content.

Usage: `ait stash drop [OPTIONS] <STASH_ID>`

Arguments:

`<STASH_ID>`

Exact active stash ID.

Options:

`--json`

Emit the complete machine-readable JSON result.

`-h, --help`

Print help

CLI · COMPLETE 1.1.1 REFERENCE

ait change - Complete Reference

Create, inspect, update, cancel, and otherwise manage the Change lifecycle within a Task.

AUDIENCE Coding agents and maintainers

<https://ait-native.dev/technical/v1.1.1/cli/reference/change/>

Operating boundary

Purpose: Create, inspect, update, cancel, and otherwise manage the Change lifecycle within a Task.

Effect class: Mixed Change reads and lifecycle mutation.

Coverage: 8 public help surfaces under `ait change`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait change
ait change create
ait change list
ait change show
ait change revert
ait change replay
ait change close
ait change publish
```

Exact emitted help

ait change

Binary description: Create, inspect, publish, or close Task-owned Changes.

```
Create, inspect, publish, or close Task-owned Changes.

Usage: ait change <COMMAND>

Commands:
  create  Create an additional Change for an existing Task.
  list    List open Changes or complete Change history locally or on a remote.
  show    Inspect one Task-owned Change without changing it.
  revert  Remove one Change's recorded delta from the current workspace.
  replay  Apply one Change's recorded delta to the current workspace.
  close   Archive one Change without finishing it.
  publish Promote one local draft Change record to a remote.
  help    Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

ait change create

Binary description: Create an additional Change for an existing Task.

```
Create an additional Change for an existing Task.

Usage: ait change create [OPTIONS] --title <TITLE> <TASK_ID>

Arguments:
  <TASK_ID> Existing Task ID that will own the new Change.

Options:
  --title <TITLE> Required title for the new Change.
  --base-line <LINE> Base Line assertion; defaults to the bound worktree target Line, then
                    the repository default Line.
  --local Create the Change locally even when Change commands default to a
          remote.
  --remote <REMOTE> Create the Change on the named remote even when Change commands
                  default to local.
  --json Emit stable machine-readable JSON.
  -h, --help Print help
```

ait change list

Binary description: List open Changes or complete Change history locally or on a remote.

```
List open Changes or complete Change history locally or on a remote.

Usage: ait change list [OPTIONS]

Options:
  --local List local Changes even when Change commands default to a remote.
  --remote <REMOTE> List Changes from the named remote even when Change commands default to
                  local.
  --all Show complete Change history instead of the bounded open view.
  --json Emit stable machine-readable JSON; the bounded view still applies
        unless --all is present.
  -h, --help Print help
```

ait change show

Binary description: Inspect one Task-owned Change without changing it.

```
Inspect one Task-owned Change without changing it.

Usage: ait change show [OPTIONS] <TASK_ID/C-##>

Arguments:
  <TASK_ID/C-##> Task-owned Change reference; a bare C-## is accepted only when it identifies
                one Change.

Options:
  --local Read the local Change even when Change commands default to a remote.
  --remote <REMOTE> Read the Change from the named remote even when Change commands default
                  to local.
  --json Emit the complete stable machine-readable Change payload.
  -h, --help Print help
```

ait change revert

Binary description: Remove one Change's recorded fork-to-revision delta from the current workspace. This does not create a Snapshot, move the current Line head, close the Change, or change remote data.

Remove one Change's recorded fork-to-revision delta from the current workspace. This does not create a Snapshot, move the current Line head, close the Change, or change remote data.

Usage: `ait change revert [OPTIONS] <TASK_ID/C-##>`

Arguments:

`<TASK_ID/C-##>`

Task-owned Change whose recorded delta will be removed from the workspace.

Options:

`--force`

Overwrite unsaved changes on paths selected by the Change delta.

`--dry-run`

Preview affected paths and overwrite risk without modifying the workspace.

`--local`

Read the Change's local history even when Change commands default to a remote.

`--remote <REMOTE>`

Read the Change's history from the named remote; only local workspace files are changed.

`--json`

Emit the complete stable machine-readable workspace plan or result.

`-h, --help`

Print help (see a summary with `'-h'`)

ait change replay

Binary description: Apply one Change's recorded fork-to-revision delta to the current Line workspace. This does not create a Snapshot, move the Line head, finish the Change, or change remote data.

Apply one Change's recorded fork-to-revision delta to the current Line workspace. This does not create a Snapshot, move the Line head, finish the Change, or change remote data.

Usage: `ait change replay [OPTIONS] <TASK_ID/C-##>`

Arguments:

`<TASK_ID/C-##>`

Task-owned Change whose recorded delta will be applied to the workspace.

Options:

`--force`

Overwrite unsaved changes on paths selected by the Change delta.

`--dry-run`

Preview affected paths and overwrite risk without modifying the workspace.

`--local`

Read the Change's local history even when Change commands default to a remote.

`--remote <REMOTE>`

Read the Change's history from the named remote; only local workspace files are changed.

`--json`

Emit the complete stable machine-readable workspace plan or result.

`-h, --help`

Print help (see a summary with `'-h'`)

ait change close

Binary description: Archive one local or remote Change without applying its code to the target Line. A successful close also attempts a bounded safe Plan repair for the owning Task.

Archive one local or remote Change without applying its code to the target Line. A successful close also attempts a bounded safe Plan repair for the owning Task.

Usage: ait change close [OPTIONS] <TASK_ID/C-##>

Arguments:

<TASK_ID/C-##>

Task-owned Change to archive without applying it to the target Line.

Options:

--local

Archive the local Change even when Change commands default to a remote.

--remote <REMOTE>

Archive the Change on the named remote even when Change commands default to local.

--json

Emit the complete stable machine-readable closeout payload.

-h, --help

Print help (see a summary with '-h')

ait change publish

Binary description: Promote one local draft Change record to the configured or named remote. This does not publish a Patchset, publish current workspace content, or finish the Change.

Promote one local draft Change record to the configured or named remote. This does not publish a Patchset, publish current workspace content, or finish the Change.

Usage: ait change publish [OPTIONS] <TASK_ID/C-##>

Arguments:

<TASK_ID/C-##>

Local Task-owned draft Change to promote.

Options:

--remote <REMOTE>

Destination remote; omitted uses the repository default remote.

--json

Emit the complete stable machine-readable publication payload.

-h, --help

Print help (see a summary with '-h')

CLI · COMPLETE 1.1.1 REFERENCE

ait snapshot - Complete Reference

Create and inspect immutable repository state, traverse its parent history, and replay or restore recorded content.

AUDIENCE Coding agents and maintainers

<https://ait-native.dev/technical/v1.1.1/cli/reference/snapshot/>

Operating boundary

Purpose: Create and inspect immutable repository state, traverse its parent history, and replay or restore recorded content.

Effect class: Create records immutable state; revert, replay, and restore-related operations change current local state as described.

Coverage: 11 public help surfaces under `ait snapshot`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait snapshot
ait snapshot create
ait snapshot list
ait snapshot show
ait snapshot diff
ait snapshot restore-lines
ait snapshot revert
ait snapshot replay
ait snapshot ancestry
ait snapshot is-ancestor
ait snapshot merge-base
```

Exact emitted help

ait snapshot

Binary description: Create, inspect, compare, and query immutable local Snapshots. Snapshot create advances the current Line head; restore-lines, revert, and replay change only workspace files and never create a Snapshot, move a Line head, or change remote data.

Create, inspect, compare, and query immutable local Snapshots. Snapshot create advances the current Line head; restore-lines, revert, and replay change only workspace files and never create a Snapshot, move a Line head, or change remote data.

Usage: ait snapshot <COMMAND>

Commands:

create	Capture the managed workspace as a new Snapshot.
list	List repository-wide local Snapshot history.
show	Inspect one immutable local Snapshot.
diff	Compare two immutable local Snapshots.
restore-lines	Preview or apply an exact line range from one Snapshot into the workspace.
revert	Remove the current head Snapshot's recorded delta from the workspace.
replay	Apply one Snapshot's recorded delta to the current Line workspace.
ancestry	Query bounded Snapshot ancestors or descendants using metadata-only DAG traversal.
is-ancestor	Test DAG ancestry; exits 0 when true, 1 when false, and 2 on lookup or storage errors.
merge-base	Find the deterministic best common Snapshot ancestor.
help	Print this message or the help of the given subcommand(s)

Options:

-h, --help
Print help (see a summary with '-h')

ait snapshot create

Binary description: Capture the managed workspace as a new immutable local Snapshot and advance the current Line head. This command does not publish remote data.

Capture the managed workspace as a new immutable local Snapshot and advance the current Line head. This command does not publish remote data.

Usage: ait snapshot create [OPTIONS]

Options:

-m, --message <MESSAGE>
Record an optional human-readable Snapshot message.

--json
Emit the compact versioned machine-readable creation payload.

--full
With --json, emit the previous complete stable creation payload.

-h, --help
Print help (see a summary with '-h')

ait snapshot list

Binary description: List local Snapshots across every Line without modifying repository state. Text and JSON output are bounded to the recent view by default; --all emits complete history.

List local Snapshots across every Line without modifying repository state. Text and JSON output are bounded to the recent view by default; --all emits complete history.

Usage: ait snapshot list [OPTIONS]

Options:

--all
Show complete Snapshot history instead of the bounded recent text or JSON view.

--json
Emit the stable machine-readable Snapshot list.

-h, --help
Print help (see a summary with '-h')

ait snapshot show

Binary description: Inspect one Snapshot resolved from an exact Snapshot ID or local AIT Tag. Default text includes identity and bounded primary-parent change evidence; --files expands the complete tree inventory. JSON is always complete.

Inspect one Snapshot resolved from an exact Snapshot ID or local AIT Tag. Default text includes identity and bounded primary-parent change evidence; --files expands the complete tree inventory. JSON is always complete.

Usage: ait snapshot show [OPTIONS] <SNAPSHOT_OR_TAG>

Arguments:

<SNAPSHOT_OR_TAG>

Exact immutable Snapshot ID or local AIT Tag to inspect.

Options:

--files

Show the complete Snapshot tree inventory in text output; JSON is always complete.

--json

Emit the complete stable machine-readable Snapshot payload.

-h, --help

Print help (see a summary with '-h')

ait snapshot diff

Binary description: Compare two Snapshots resolved from exact Snapshot IDs or local AIT Tags. Structural file changes are reported by default; --include-text adds bounded text diffs for eligible modified files. This command is read-only.

Compare two Snapshots resolved from exact Snapshot IDs or local AIT Tags. Structural file changes are reported by default; --include-text adds bounded text diffs for eligible modified files. This command is read-only.

Usage: ait snapshot diff [OPTIONS] <OLD_SNAPSHOT_OR_TAG> <NEW_SNAPSHOT_OR_TAG>

Arguments:

<OLD_SNAPSHOT_OR_TAG>

Exact Snapshot ID or local AIT Tag for the old side.

<NEW_SNAPSHOT_OR_TAG>

Exact Snapshot ID or local AIT Tag for the new side.

Options:

--include-text

Include bounded line-oriented text diffs for eligible modified files.

--max-bytes <BYTES>

Maximum old or new file size allowed for text diff generation; requires --include-text.

[default: 128000]

--json

Emit the complete stable machine-readable diff payload.

-h, --help

Print help (see a summary with '-h')

ait snapshot restore-lines

Binary description: Read one exact immutable Snapshot ID and preview replacement of the same existing regular workspace path at one positive 1-based line or inclusive range. Only --yes applies the selected lines; the command never creates a Snapshot or moves a Line head.

Read one exact immutable Snapshot ID and preview replacement of the same existing regular workspace path at one positive 1-based line or inclusive range. Only --yes applies the selected lines; the command never creates a Snapshot or moves a Line head.

Usage: ait snapshot restore-lines <SNAPSHOT_ID> <PATH> (--line <N> | --start <N> --end <N>) [--yes] [--json]

Arguments:

- <SNAPSHOT_ID>
Exact immutable Snapshot ID to read line content from.
- <PATH>
Existing regular workspace file to update.

Options:

- line <LINE>
Select one 1-based workspace line.
- start <START_LINE>
Start of one inclusive 1-based line range.
- end <END_LINE>
End of one inclusive 1-based line range.
- yes
Apply the selected Snapshot lines; omission is always a read-only preview.
- json
Emit the complete stable machine-readable preview or apply result.
- h, --help
Print help (see a summary with '-h')

ait snapshot revert

Binary description: Remove one Line Snapshot's primary-parent-to-revision delta from the current workspace. The requested Snapshot must be the current Line head. This does not create a Snapshot, move the current Line head, or mutate remote state.

Remove one Line Snapshot's primary-parent-to-revision delta from the current workspace. The requested Snapshot must be the current Line head. This does not create a Snapshot, move the current Line head, or mutate remote state.

Usage: ait snapshot revert [OPTIONS] <SNAPSHOT_OR_TAG>

Arguments:

- <SNAPSHOT_OR_TAG>
Current Line head Snapshot, identified by exact ID or local AIT Tag.

Options:

- force
Overwrite unsaved changes on paths selected by the Snapshot delta.
- dry-run
Preview affected paths and overwrite risk without modifying the workspace.
- json
Emit the complete stable machine-readable workspace plan or result.
- h, --help
Print help (see a summary with '-h')

ait snapshot replay

Binary description: Apply one non-root Line Snapshot's primary-parent-to-revision delta to the current Line workspace. This does not create a Snapshot, move the current Line head, or mutate remote state.

Apply one non-root Line Snapshot's primary-parent-to-revision delta to the current Line workspace. This does not create a Snapshot, move the current Line head, or mutate remote state.

Usage: `ait snapshot replay [OPTIONS] <SNAPSHOT_OR_TAG>`

Arguments:

`<SNAPSHOT_OR_TAG>`

Non-root Snapshot whose recorded delta will be applied to the workspace.

Options:

`--force`

Overwrite unsaved changes on paths selected by the Snapshot delta.

`--dry-run`

Preview affected paths and overwrite risk without modifying the workspace.

`--json`

Emit the complete stable machine-readable workspace plan or result.

`-h, --help`

Print help (see a summary with `'-h'`)

ait snapshot ancestry

Binary description: Query deterministic topological Snapshot ancestry resolved from an exact Snapshot ID or local AIT Tag. Ancestors across all parents are the default; direction, first-parent traversal, depth, and result bounds are explicit. The query Snapshot itself is excluded.

Query deterministic topological Snapshot ancestry resolved from an exact Snapshot ID or local AIT Tag. Ancestors across all parents are the default; direction, first-parent traversal, depth, and result bounds are explicit. The query Snapshot itself is excluded.

Usage: `ait snapshot ancestry [OPTIONS] <SNAPSHOT_OR_TAG>`

Arguments:

`<SNAPSHOT_OR_TAG>`

Exact Snapshot ID or local AIT Tag at the query boundary.

Options:

`--ancestors`

Traverse ancestors; this is the default direction.

`--descendants`

Traverse descendants instead of ancestors.

`--first-parent`

Follow only ordinal-zero parent edges instead of all parents.

`--max-depth <DEPTH>`

Stop traversal after this positive edge depth.

[default: 10000]

`--limit <COUNT>`

Admit at most this positive number of results.

[default: 10000]

`--all`

Show every result allowed by `--limit` in text output instead of the nearest 20; JSON already includes all of them.

`--json`

Emit the complete stable machine-readable bounded query payload.

`-h, --help`

Print help (see a summary with `'-h'`)

ait snapshot is-ancestor

Binary description: Test whether one Snapshot is reachable as an ancestor of another across the complete parent DAG. Each input accepts an exact Snapshot ID or local AIT Tag. The command exits 0 when true, 1 when false, and 2 on lookup or storage errors.

Test whether one Snapshot is reachable as an ancestor of another across the complete parent DAG. Each input accepts an exact Snapshot ID or local AIT Tag. The command exits 0 when true, 1 when false, and 2 on lookup or storage errors.

Usage: ait snapshot is-ancestor [OPTIONS] <OLDER_SNAPSHOT_OR_TAG> <NEWER_SNAPSHOT_OR_TAG>

Arguments:

<OLDER_SNAPSHOT_OR_TAG>

Candidate ancestor, identified by exact Snapshot ID or local AIT Tag.

<NEWER_SNAPSHOT_OR_TAG>

Candidate descendant, identified by exact Snapshot ID or local AIT Tag.

Options:

--json

Emit the stable machine-readable ancestry decision and distance.

-h, --help

Print help (see a summary with '-h')

ait snapshot merge-base

Binary description: Find the best common ancestor of two Snapshots resolved from exact Snapshot IDs or local AIT Tags. The default emits one deterministic best base; --all emits every equally best base. The command exits 0 when a base exists, 1 when none exists, and 2 on lookup or storage errors.

Find the best common ancestor of two Snapshots resolved from exact Snapshot IDs or local AIT Tags. The default emits one deterministic best base; --all emits every equally best base. The command exits 0 when a base exists, 1 when none exists, and 2 on lookup or storage errors.

Usage: ait snapshot merge-base [OPTIONS] <LEFT_SNAPSHOT_OR_TAG> <RIGHT_SNAPSHOT_OR_TAG>

Arguments:

<LEFT_SNAPSHOT_OR_TAG>

Left Snapshot, identified by exact ID or local AIT Tag.

<RIGHT_SNAPSHOT_OR_TAG>

Right Snapshot, identified by exact ID or local AIT Tag.

Options:

--all

Emit every equally best common ancestor in deterministic order.

--json

Emit the complete stable machine-readable merge-base result.

-h, --help

Print help (see a summary with '-h')

CLI · COMPLETE 1.1.1 REFERENCE

ait commit - Complete Reference

Create an AIT Snapshot through a Git-friendly commit-named entry point.

AUDIENCE Developers and coding agents

<https://ait-native.dev/technical/v1.1.1/cli/reference/commit/>

Operating boundary

Purpose: Create an AIT Snapshot through a Git-friendly commit-named entry point.

Effect class: Records immutable local repository state using the same Snapshot authority as snapshot create.

Coverage: 1 public help surface under `ait commit`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait commit
```

Exact emitted help

`ait commit`

Binary description: Git-friendly alias for `ait snapshot create`. Capture the managed workspace as a new immutable local Snapshot and advance the current Line head. The result remains an AIT Snapshot, uses the same output and safety checks, and does not publish remote data.

```
Git-friendly alias for `ait snapshot create`. Capture the managed workspace as a new immutable
local Snapshot and advance the current Line head. The result remains an AIT Snapshot, uses the
same output and safety checks, and does not publish remote data.
```

```
Usage: ait commit [OPTIONS]
```

```
Options:
```

```
-m, --message <MESSAGE>
```

```
    Record an optional human-readable Snapshot message.
```

```
--json
```

```
    Emit the compact versioned machine-readable creation payload.
```

```
--full
```

```
    With --json, emit the previous complete stable creation payload.
```

```
-h, --help
```

```
    Print help (see a summary with '-h')
```

CLI · COMPLETE 1.1.1 REFERENCE

ait tag - Complete Reference

Create, inspect, list, and remove local AIT Tag references to recorded state.

AUDIENCE Developers and release operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/tag/>

Operating boundary

Purpose: Create, inspect, list, and remove local AIT Tag references to recorded state.

Effect class: Tag-reference mutation with read-only list and show operations.

Coverage: 5 public help surfaces under `ait tag`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait tag
ait tag create
ait tag list
ait tag show
ait tag delete
```

Exact emitted help

`ait tag`

Binary description: Create, inspect, list, and delete local-only AIT Tags that name exact Snapshots. An existing Tag binding cannot be retargeted, and deleting a Tag never deletes its Snapshot.

```
Create, inspect, list, and delete local-only AIT Tags that name exact Snapshots. An existing Tag
binding cannot be retargeted, and deleting a Tag never deletes its Snapshot.
```

```
Usage: ait tag <COMMAND>
```

Commands:

```
create  Create a local Tag; existing names are rejected.
list    List all local Tags without changing repository state.
show    Show one exact local Tag without changing repository state.
delete  Delete only one local Tag reference.
help    Print this message or the help of the given subcommand(s)
```

Options:

```
-h, --help
        Print help (see a summary with '-h')
```

`ait tag create`

Binary description: Create one local AIT Tag for an exact existing Snapshot. When `--snapshot` is omitted, the current Line head is used. An existing Tag name is always rejected and cannot be replaced or moved.

Create one local AIT Tag for an exact existing Snapshot. When `--snapshot` is omitted, the current Line head is used. An existing Tag name is always rejected and cannot be replaced or moved.

Usage: `ait tag create [OPTIONS] --message <MESSAGE> <NAME>`

Arguments:

`<NAME>`

New local Tag name; an existing name is rejected.

Options:

`--snapshot <SNAPSHOT_ID>`

Exact existing local Snapshot ID; defaults to the current Line head.

`--message <MESSAGE>`

Required non-empty, single-line reason for creating the Tag.

`--json`

Emit the created Tag as stable JSON.

`-h, --help`

Print help (see a summary with `'-h'`)

ait tag list

Binary description: List all local Tags without changing repository state.

List all local Tags without changing repository state.

Usage: `ait tag list [OPTIONS]`

Options:

`--json` Emit the complete local Tag list as stable JSON.

`-h, --help` Print help

ait tag show

Binary description: Show one exact local Tag without changing repository state.

Show one exact local Tag without changing repository state.

Usage: `ait tag show [OPTIONS] <NAME>`

Arguments:

`<NAME>` Exact local Tag name to inspect.

Options:

`--json` Emit the Tag record as stable JSON.

`-h, --help` Print help

ait tag delete

Binary description: Delete one exact local AIT Tag reference. The referenced Snapshot, Line heads, and workspace content remain unchanged.

Delete one exact local AIT Tag reference. The referenced Snapshot, Line heads, and workspace content remain unchanged.

Usage: `ait tag delete [OPTIONS] <NAME>`

Arguments:

`<NAME>`

Exact local Tag name to delete.

Options:

`--json`

Emit the deleted Tag record as stable JSON.

`-h, --help`

Print help (see a summary with `'-h'`)

CLI · COMPLETE 1.1.1 REFERENCE

ait patchset - Complete Reference

Publish and inspect Change candidates, select the current candidate, and manage its CI request state.

AUDIENCE Coding agents and CI operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/patchset/>

Operating boundary

Purpose: Publish and inspect Change candidates, select the current candidate, and manage its CI request state.

Effect class: Remote candidate, selection, and CI-state mutation with read-only inspection operations.

Coverage: 7 public help surfaces under `ait patchset`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait patchset
ait patchset publish
ait patchset list
ait patchset show
ait patchset select
ait patchset ci-status
ait patchset rerun-ci
```

Exact emitted help

ait patchset

Binary description: Publish the current local Line head as a remote Change revision, inspect or select specific published Patchsets, and read or manually rerun their remote CI. Published Patchsets exist only on remotes; `--remote` selects a configured remote and there is no local Patchset mode.

```
Publish the current local Line head as a remote Change revision, inspect or select specific
published Patchsets, and read or manually rerun their remote CI. Published Patchsets exist
only on remotes; --remote selects a configured remote and there is no local Patchset mode.
```

```
Usage: ait patchset <COMMAND>
```

Commands:

<code>publish</code>	Publish the current Line head as a new remote Patchset.
<code>list</code>	List the published Patchsets owned by one remote Change without modifying it.
<code>show</code>	Show one exact published remote Patchset without modifying it.
<code>select</code>	Select one specific Patchset on its owning remote Change.
<code>ci-status</code>	Read CI state for one exact remote Patchset.
<code>rerun-ci</code>	Queue a manual CI rerun for one exact remote Patchset.
<code>help</code>	Print this message or the help of the given subcommand(s)

Options:

<code>-h, --help</code>	Print help (see a summary with <code>'-h'</code>)
-------------------------	--

ait patchset publish

Binary description: Publish the current local Line head as the next Patchset for `CHANGE_ID` after validating the bound worktree and synchronizing its revision Snapshot to the selected remote. `--summary` is required; `--author-mode` overrides the configured provenance mode for this publication only.

Publish the current local Line head as the next Patchset for `CHANGE_ID` after validating the bound worktree and synchronizing its revision Snapshot to the selected remote. `--summary` is required; `--author-mode` overrides the configured provenance mode for this publication only.

Usage: `ait patchset publish [OPTIONS] --summary <SUMMARY> <CHANGE_ID>`

Arguments:

`<CHANGE_ID>`

Exact remote Change ID that will own the new Patchset.

Options:

`--summary <SUMMARY>`

Required human-readable summary of this published revision.

`--author-mode <MODE>`

Override provenance mode for this publication; otherwise use configured `default_author_mode`.

[possible values: `human_only`, `human_with_ai_assist`, `ai_with_human_review`, `ai_only_experimental`]

`--remote <REMOTE>`

Configured remote name; defaults to the repository's default remote.

`--json`

Emit the complete publication result as stable JSON.

`-h, --help`

Print help (see a summary with `'-h'`)

ait patchset list

Binary description: List the published Patchsets owned by one remote Change without modifying it.

List the published Patchsets owned by one remote Change without modifying it.

Usage: `ait patchset list [OPTIONS] <CHANGE_ID>`

Arguments:

`<CHANGE_ID>` Exact remote Change ID whose published Patchsets will be listed.

Options:

`--remote <REMOTE>` Configured remote name; defaults to the repository's default remote.

`--json` Emit the complete Patchset list as stable JSON.

`-h, --help`

Print help

ait patchset show

Binary description: Show one exact published remote Patchset without modifying it.

Show one exact published remote Patchset without modifying it.

Usage: `ait patchset show [OPTIONS] <PATCHSET_ID>`

Arguments:

`<PATCHSET_ID>` Complete published Patchset ID; bare numeric ordinals are rejected.

Options:

`--remote <REMOTE>` Configured remote name; defaults to the repository's default remote.

`--json` Emit the complete Patchset record as stable JSON.

`-h, --help`

Print help

ait patchset select

Binary description: Read the requested Patchset, find its owning remote Change, then make that Patchset the Change's selected revision. The owning Change cannot be supplied or overridden by the caller.

Read the requested Patchset, find its owning remote Change, then make that Patchset the Change's selected revision. The owning Change cannot be supplied or overridden by the caller.

Usage: `ait patchset select [OPTIONS] <PATCHSET_ID>`

Arguments:

`<PATCHSET_ID>`
Complete published Patchset ID; its owning Change is derived remotely.

Options:

`--remote <REMOTE>`
Configured remote name; defaults to the repository's default remote.

`--json`
Emit the complete selection result as stable JSON.

`-h, --help`
Print help (see a summary with `'-h'`)

ait patchset ci-status

Binary description: Read current CI readiness and the 10 most recent CI jobs for one published Patchset without changing remote data.

Read current CI readiness and the 10 most recent CI jobs for one published Patchset without changing remote data.

Usage: `ait patchset ci-status [OPTIONS] <PATCHSET_ID>`

Arguments:

`<PATCHSET_ID>`
Complete published Patchset ID; bare numeric ordinals are rejected.

Options:

`--remote <REMOTE>`
Configured remote name; defaults to the repository's default remote.

`--json`
Emit the complete CI status and fixed recent-job window as stable JSON.

`-h, --help`
Print help (see a summary with `'-h'`)

ait patchset rerun-ci

Binary description: Queue CI for one exact published Patchset using the fixed trigger `manual_rerun`. Runner selection and execution profiles remain server policy and cannot be overridden here.

Queue CI for one exact published Patchset using the fixed trigger `manual_rerun`. Runner selection and execution profiles remain server policy and cannot be overridden here.

Usage: `ait patchset rerun-ci [OPTIONS] <PATCHSET_ID>`

Arguments:

`<PATCHSET_ID>`
Complete published Patchset ID to enqueue for a manual rerun.

Options:

`--remote <REMOTE>`
Configured remote name; defaults to the repository's default remote.

`--json`
Emit the complete queued-run result as stable JSON.

`-h, --help`
Print help (see a summary with `'-h'`)

CLI · COMPLETE 1.1.1 REFERENCE

ait review - Complete Reference

Request, inspect, and record review decisions for a selected Patchset.

AUDIENCE Reviewers, coding agents, and maintainers

<https://ait-native.dev/technical/v1.1.1/cli/reference/review/>

Operating boundary

Purpose: Request, inspect, and record review decisions for a selected Patchset.

Effect class: Remote review workflow mutation and inspection.

Coverage: 16 public help surfaces under `ait review`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait review
ait review show
ait review team
ait review team request
ait review team approve
ait review team request-changes
ait review team comment
ait review team defer
ait review task
ait review task approve
ait review task request-changes
ait review task comment
ait review task defer
ait review code
ait review code submit
ait review code template
```

Exact emitted help

`ait review`

Binary description: Inspect remote Review state and record team, human Task, or AI code-review evidence.

```
Inspect remote Review state and record team, human Task, or AI code-review evidence.

Usage: ait review <COMMAND>

Commands:
  show  Show compact remote Review state for one Change.
  team  Manage team governance review; available only in team_remote mode.
  task  Record human functional Task review using only configured user_name.
  code  Generate or submit structured AI code-review evidence attributed to the executing app.
  help  Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

`ait review show`

Binary description: Show compact remote Review state for one Change.

```

Show compact remote Review state for one Change.

Usage: ait review show [OPTIONS] <CHANGE_ID>

Arguments:
  <CHANGE_ID>  Remote Change ID or supported Change reference to inspect.

Options:
  --remote <REMOTE>  Configured remote name; defaults to the repository's default remote.
  --json             Emit the compact stable machine-readable Review result.
  -h, --help         Print help

```

ait review team

Binary description: Manage team governance review; available only in team_remote mode.

```

Manage team governance review; available only in team_remote mode.

Usage: ait review team <COMMAND>

Commands:
  request
  approve
  request-changes
  comment
  defer
  help          Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help

```

ait review team request

Reference description: Request team review from one or more exact reviewer groups.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```

Usage: ait review team request [OPTIONS] --group <REVIEWER_GROUPS> <CHANGE_ID>

Arguments:
  <CHANGE_ID>

Options:
  --group <REVIEWER_GROUPS>
  --patchset <PATCHSET_ID>
  --note <NOTE>
  --remote <REMOTE>
  --json
  -h, --help          Print help

```

ait review team approve

Reference description: Record a team-review approval for the selected Change.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```

Usage: ait review team approve [OPTIONS] <CHANGE_ID>

Arguments:
  <CHANGE_ID>

Options:
  --reviewer <REVIEWER>
  --patchset <PATCHSET_ID>
  --message <MESSAGE>
  --remote <REMOTE>
  --json
  -h, --help          Print help

```

ait review team request-changes

Reference description: Record a team-review request for changes on the selected Change.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```
Usage: ait review team request-changes [OPTIONS] <CHANGE_ID>

Arguments:
  <CHANGE_ID>

Options:
  --reviewer <REVIEWER>
  --patchset <PATCHSET_ID>
  --message <MESSAGE>
  --remote <REMOTE>
  --json
  -h, --help                Print help
```

ait review team comment

Reference description: Record a non-decisional team-review comment on the selected Change.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```
Usage: ait review team comment [OPTIONS] <CHANGE_ID>

Arguments:
  <CHANGE_ID>

Options:
  --reviewer <REVIEWER>
  --patchset <PATCHSET_ID>
  --message <MESSAGE>
  --remote <REMOTE>
  --json
  -h, --help                Print help
```

ait review team defer

Reference description: Defer the current team-review decision for the selected Change.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```
Usage: ait review team defer [OPTIONS] <CHANGE_ID>

Arguments:
  <CHANGE_ID>

Options:
  --reviewer <REVIEWER>
  --patchset <PATCHSET_ID>
  --message <MESSAGE>
  --remote <REMOTE>
  --json
  -h, --help                Print help
```

ait review task

Binary description: Record human functional Task review using only configured user_name.

```
Record human functional Task review using only configured user_name.

Usage: ait review task <COMMAND>

Commands:
  approve          Approve one specific Patchset after functional validation;
                   task_review=required only.
  request-changes
  comment
  defer
  help             Print this message or the help of the given subcommand(s)

Options:
  -h, --help      Print help
```

ait review task approve

Binary description: Approve one specific Patchset after functional validation; task_review=required only.

```
Approve one specific Patchset after functional validation; task_review=required only.

Usage: ait review task approve [OPTIONS] --patchset <PATCHSET_ID> --message <MESSAGE>
      <CHANGE_ID>

Arguments:
  <CHANGE_ID>  Remote Change that owns the functionally validated Patchset.

Options:
  --patchset <PATCHSET_ID> Complete published Patchset ID whose functionality was
                           validated; numeric Repository references are rejected.
  --message <MESSAGE>      Non-empty functional-validation evidence from the configured
                           user_name.
  --remote <REMOTE>        Configured remote name; defaults to the repository's default
                           remote.
  --json                  Emit the recorded Task approval as stable JSON.
  -h, --help              Print help
```

ait review task request-changes

Reference description: Record a task-scope request for changes on the selected Change.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```
Usage: ait review task request-changes [OPTIONS] <CHANGE_ID>

Arguments:
  <CHANGE_ID>

Options:
  --patchset <PATCHSET_ID>
  --message <MESSAGE>
  --remote <REMOTE>
  --json
  -h, --help              Print help
```

ait review task comment

Reference description: Record a non-decisional task-scope review comment on the selected Change.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```
Usage: ait review task comment [OPTIONS] <CHANGE_ID>
```

```
Arguments:
  <CHANGE_ID>
```

```
Options:
  --patchset <PATCHSET_ID>
  --message <MESSAGE>
  --remote <REMOTE>
  --json
  -h, --help                Print help
```

ait review task defer

Reference description: Defer the current task-scope review decision for the selected Change.

The 1.1.1 binary emits no separate prose description for this surface; the sentence above is editorial context. The complete parser-provided usage and options follow unchanged.

```
Usage: ait review task defer [OPTIONS] <CHANGE_ID>
```

```
Arguments:
  <CHANGE_ID>
```

```
Options:
  --patchset <PATCHSET_ID>
  --message <MESSAGE>
  --remote <REMOTE>
  --json
  -h, --help                Print help
```

ait review code

Binary description: Generate or submit structured AI code-review evidence attributed to the executing app.

Generate or submit structured AI code-review evidence attributed to the executing app.

```
Usage: ait review code <COMMAND>
```

```
Commands:
  submit    Submit the implicit pass outcome for one exact, fully inspected Patchset.
  template  Print the required structured AI code-review summary template locally.
  help      Print this message or the help of the given subcommand(s)
```

```
Options:
  -h, --help  Print help
```

ait review code submit

Binary description: Submit the implicit pass outcome for one exact, fully inspected Patchset.

Submit the implicit pass outcome for one exact, fully inspected Patchset.

```
Usage: ait review code submit [OPTIONS] --patchset <PATCHSET_ID> --message <MESSAGE> <CHANGE_ID>
```

```
Arguments:
  <CHANGE_ID>  Remote Change that owns the exact reviewed Patchset.
```

```
Options:
  --patchset <PATCHSET_ID>  Complete published Patchset ID reviewed by the executing AI app;
                             numeric Repository references are rejected.
  --message <MESSAGE>        Structured Reviewed files, Findings, Risks, Tests, and pass
                             Recommendation summary for this specific Patchset.
  --remote <REMOTE>          Configured remote name; defaults to the repository's default
                             remote.
  --json                     Emit both code and Task review-lane results as stable JSON.
  -h, --help                Print help
```

ait review code template

Binary description: Print the required structured AI code-review summary template locally.

Print the required structured AI code-review summary template locally.

Usage: ait review code template [OPTIONS]

Options:

--style <STYLE>	Template layout: inline or numbered. [default: numbered] [possible values: inline, numbered]
--json	Emit the template metadata as stable JSON.
-h, --help	Print help

CLI · COMPLETE 1.1.1 REFERENCE

ait attest - Complete Reference

Create and inspect typed evidence attached to a selected Patchset.

AUDIENCE Coding agents and evidence producers

<https://ait-native.dev/technical/v1.1.1/cli/reference/attest/>

Operating boundary

Purpose: Create and inspect typed evidence attached to a selected Patchset.

Effect class: Remote evidence mutation and inspection.

Coverage: 3 public help surfaces under `ait attest`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait attest
ait attest put
ait attest show
```

Exact emitted help

`ait attest`

Binary description: Record and inspect evidence for remote Patchsets.

```
Record and inspect evidence for remote Patchsets.

Usage: ait attest <COMMAND>

Commands:
  put   Record test, lint, security, license, and authorship evidence for a remote Patchset.
  show  Show recorded evidence for one remote Patchset.
  help  Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

`ait attest put`

Binary description: Record test, lint, security, license, and authorship evidence for a remote Patchset.

```
Record test, lint, security, license, and authorship evidence for a remote Patchset.

Usage: ait attest put [OPTIONS] [PATCHSET_ID]

Arguments:
  [PATCHSET_ID]

Options:
  --change <CHANGE>
  --tests <TESTS>
  --lint <LINT>
  --security <SECURITY>
  --license <LICENSE>
  --author-mode <AUTHOR_MODE>
  --model <MODEL>
  --remote <REMOTE>
  --json
  -h, --help  Print help
```

ait attest show

Binary description: Show recorded evidence for one remote Patchset.

Show recorded evidence for one remote Patchset.

Usage: ait attest show [OPTIONS] <PATCHSET_ID>

Arguments:

<PATCHSET_ID>

Options:

--remote <REMOTE>

--json

-h, --help Print help

CLI · COMPLETE 1.1.1 REFERENCE

ait policy - Complete Reference

Evaluate, inspect, and explicitly waive supported policy checks for a selected Patchset.

AUDIENCE Maintainers and policy operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/policy/>

Operating boundary

Purpose: Evaluate, inspect, and explicitly waive supported policy checks for a selected Patchset.

Effect class: Evaluation and inspection plus an explicit reason-bearing policy-waiver mutation.

Coverage: 4 public help surfaces under `ait policy`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait policy
ait policy eval
ait policy show
ait policy waive
```

Exact emitted help

ait policy

Binary description: Check, inspect, or waive remote Patchset requirements.

```
Check, inspect, or waive remote Patchset requirements.

Usage: ait policy <COMMAND>

Commands:
  eval  Check the configured Policy for a selected remote Patchset.
  show  Show the recorded Policy result for a remote Patchset.
  waive Request a reasoned Policy waiver for a remote Patchset.
  help  Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

ait policy eval

Binary description: Check the configured Policy for a selected remote Patchset.

```
Check the configured Policy for a selected remote Patchset.

Usage: ait policy eval [OPTIONS] <PATCHSET_ID>

Arguments:
  <PATCHSET_ID>

Options:
  --remote <REMOTE>
  --json
  -h, --help          Print help
```

ait policy show

Binary description: Show the recorded Policy result for a remote Patchset.

Show the recorded Policy result for a remote Patchset.

Usage: ait policy show [OPTIONS] <PATCHSET_ID>

Arguments:
 <PATCHSET_ID>

Options:
 --remote <REMOTE>
 --json
 -h, --help Print help

ait policy waive

Binary description: Request a reasoned Policy waiver for a remote Patchset.

Request a reasoned Policy waiver for a remote Patchset.

Usage: ait policy waive [OPTIONS] --rule <RULE_NAME> --reason <REASON> <PATCHSET_ID>

Arguments:
 <PATCHSET_ID>

Options:
 --rule <RULE_NAME>
 --reason <REASON>
 --expires-at <EXPIRES_AT>
 --remote <REMOTE>
 --json
 -h, --help Print help

CLI · COMPLETE 1.1.1 REFERENCE

ait worktree - Complete Reference

Inspect, diagnose, recover, relocate, and clean AIT-owned Task worktrees and their metadata.

AUDIENCE Coding agents and repository operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/worktree/>

Operating boundary

Purpose: Inspect, diagnose, recover, relocate, and clean AIT-owned Task worktrees and their metadata.

Effect class: Mixed read and local recovery/cleanup surface; exact ownership checks govern mutations.

Coverage: 16 public help surfaces under `ait worktree`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait worktree
ait worktree status
ait worktree restore
ait worktree show
ait worktree path
ait worktree doctor
ait worktree cleanup-candidates
ait worktree cleanup
ait worktree prune-stale
ait worktree list
ait worktree sync
ait worktree recreate
ait worktree recover-task
ait worktree restore-owned-head
ait worktree rebase
ait worktree remove
```

Exact emitted help

ait worktree

Binary description: Inspect and maintain isolated worktrees that share repository .ait data while carrying their own checked-out files, current Line, and optional Task/Change binding. Normal Task worktrees are created by task start. Cleanup, prune-stale, and remove require `--yes` when applied; use `--dry-run` to preview destructive removal.

Inspect and maintain isolated worktrees that share repository .ait data while carrying their own checked-out files, current Line, and optional Task/Change binding. Normal Task worktrees are created by task start. Cleanup, prune-stale, and remove require --yes when applied; use --dry-run to preview destructive removal.

Usage: ait worktree <COMMAND>

Commands:

status	Compare one checkout with a Line head or specific Snapshot.
restore	Restore all or selected checkout content from a Line head or Snapshot.
show	Show one registered worktree and its refreshed live status.
path	Print one worktree path or a shell command that enters it. [alias: open]
doctor	Summarize registered worktree health and cleanup classification.
cleanup-candidates	Inspect policy-selected cleanup candidates without removing them.
cleanup	Preview or remove policy-selected safe worktrees.
prune-stale	Preview or prune missing and detached worktree registrations.
list	List every registered worktree.
sync	Synchronize one or all worktrees to their selected Line heads.
recreate	Recreate a missing registered Task worktree.
recover-task	Recover a local authoring worktree for an existing remote Task and Change.
restore-owned-head	Restore the last contiguous Snapshot head owned by a bound Task worktree.
rebase	Replay a worktree's feature delta onto a target base Line.
remove	Preview or explicitly remove registered worktrees.
help	Print this message or the help of the given subcommand(s)

Options:

-h, --help
Print help (see a summary with '-h')

ait worktree status

Binary description: Report modified, missing, and untracked paths in the selected checkout without restoring content or moving a Line head. NAME selects a registered worktree; when omitted, inspect the current checkout. --snapshot and --line are mutually exclusive, and omission of both compares with the current Line head.

Report modified, missing, and untracked paths in the selected checkout without restoring content or moving a Line head. NAME selects a registered worktree; when omitted, inspect the current checkout. --snapshot and --line are mutually exclusive, and omission of both compares with the current Line head.

Usage: ait worktree status [OPTIONS] [NAME]

Arguments:

[NAME]
Registered worktree name; omit it to inspect the current checkout

Options:

--snapshot <SNAPSHOT_ID>
Compare with this exact local Snapshot instead of a Line head

--line <LINE_NAME>
Compare with this local Line head instead of the current Line head

--verbose
Show baseline and workspace-root detail in text output

--json
Emit the complete stable machine-readable JSON payload

-h, --help
Print help (see a summary with '-h')

ait worktree restore

Binary description: Restore the selected registered worktree, or the current checkout when NAME is omitted. With no source option, restore the whole checkout from its current Line head. --snapshot and --line are mutually exclusive. Every --path is an exact workspace-relative path and requires an explicit --snapshot or --line; selected-path restore does not switch the current Line. Unsaved changes require --force, while --dry-run only reports the restore plan.

Restore the selected registered worktree, or the current checkout when NAME is omitted. With no source option, restore the whole checkout from its current Line head. --snapshot and --line are mutually exclusive. Every --path is an exact workspace-relative path and requires an explicit --snapshot or --line; selected-path restore does not switch the current Line. Unsaved changes require --force, while --dry-run only reports the restore plan.

Usage: ait worktree restore [OPTIONS] [NAME]

Arguments:

[NAME]
Registered worktree name; omit it to restore the current checkout

Options:

--snapshot <SNAPSHOT_ID>
Restore from this exact local Snapshot; mutually exclusive with --line

--line <LINE_NAME>
Restore from this local Line head; a whole-checkout restore also switches the current Line

--path <PATH>
Restore one exact workspace-relative path; repeat for more paths and supply --snapshot or --line

--force
Overwrite unsaved changes in the selected files

--dry-run
Report writes, removals, and overwritten changes without restoring content

--json
Emit the complete stable machine-readable JSON payload

-h, --help
Print help (see a summary with '-h')

ait worktree show

Binary description: Show the registered and current Line, checked-out Snapshot, Task/Change binding, cleanup classification, and rebase or merge state for one worktree. NAME may be omitted only when the current checkout belongs to a registered worktree.

Show the registered and current Line, checked-out Snapshot, Task/Change binding, cleanup classification, and rebase or merge state for one worktree. NAME may be omitted only when the current checkout belongs to a registered worktree.

Usage: ait worktree show [OPTIONS] [NAME]

Arguments:

[NAME]
Registered worktree name; omit it to use the current runtime worktree binding

Options:

--json
Emit the complete stable machine-readable JSON payload

-h, --help
Print help (see a summary with '-h')

ait worktree path

Binary description: Print the managed alias path when available, otherwise the registered physical path. This command refreshes live status and records worktree use. --shell prints a command that changes directory and exports managed Cargo paths when enabled; the visible alias **open** has identical print-only behavior.

Print the managed alias path when available, otherwise the registered physical path. This command refreshes live status and records worktree use. `--shell` prints a command that changes directory and exports managed Cargo paths when enabled; the visible alias ``open`` has identical print-only behavior.

Usage: `ait worktree path [OPTIONS] [NAME]`

Arguments:

`[NAME]`

Registered worktree name; omit it to use the current runtime worktree binding

Options:

`--shell`

Print a shell command that enters the worktree and exports managed Cargo paths

`--json`

Emit machine-readable path, command, status, and managed runtime fields

`-h, --help`

Print help (see a summary with `'-h'`)

ait worktree doctor

Binary description: Summarize current, clean, dirty, missing, detached, protected, and cleanup-candidate worktrees. The default uses metadata and cached status without a full content scan. `--refresh` verifies live content and may refresh derived runtime layout and status caches.

Summarize current, clean, dirty, missing, detached, protected, and cleanup-candidate worktrees. The default uses metadata and cached status without a full content scan. `--refresh` verifies live content and may refresh derived runtime layout and status caches.

Usage: `ait worktree doctor [OPTIONS]`

Options:

`--refresh`

Verify each worktree's live content and refresh derived runtime status before reporting

`--json`

Emit the complete stable machine-readable JSON payload

`-h, --help`

Print help (see a summary with `'-h'`)

ait worktree cleanup-candidates

Binary description: Refresh and classify registered worktrees without removing paths or registrations. Filter by cleanup policy or idle age, opt clean manual-only worktrees into review, and optionally include protected rows with their exact reasons. Missing or detached registrations remain stale rather than cleanup candidates.

Refresh and classify registered worktrees without removing paths or registrations. Filter by cleanup policy or idle age, opt clean manual-only worktrees into review, and optionally include protected rows with their exact reasons. Missing or detached registrations remain stale rather than cleanup candidates.

Usage: `ait worktree cleanup-candidates [OPTIONS]`

Options:

`--older-than <DURATION>`
 Idle threshold for `after_idle` policy, written as integer days, hours, or minutes (for example 7d, 12h, or 30m)

 [default: 7d]

`--policy <POLICY>`
 Filter by cleanup policy: `manual_only`, `after_remote_land`, `after_task_complete`, `after_idle`, or `never`

`--allow-manual-only`
 Classify otherwise-safe clean `manual_only` worktrees as explicit cleanup candidates

`--include-protected`
 Include protected worktree rows and their reasons without making them removable

`--json`
 Emit the complete stable machine-readable JSON payload

`-h, --help`
 Print help (see a summary with `'-h'`)

ait worktree cleanup

Binary description: Select safe cleanup candidates using the same policy rules as `cleanup-candidates`. `--dry-run` previews the exact ordered removal plan without requiring confirmation. Applied cleanup deletes selected worktree paths, managed aliases, registrations, and managed Task Cargo build caches and therefore requires `--yes`.

Select safe cleanup candidates using the same policy rules as `cleanup-candidates`. `--dry-run` previews the exact ordered removal plan without requiring confirmation. Applied cleanup deletes selected worktree paths, managed aliases, registrations, and managed Task Cargo build caches and therefore requires `--yes`.

Usage: `ait worktree cleanup [OPTIONS]`

Options:

`--older-than <DURATION>`
 Idle threshold for `after_idle` policy, written as integer days, hours, or minutes (for example 7d, 12h, or 30m)

 [default: 7d]

`--policy <POLICY>`
 Filter by cleanup policy: `manual_only`, `after_remote_land`, `after_task_complete`, `after_idle`, or `never`

`--allow-manual-only`
 Permit otherwise-safe clean `manual_only` worktrees to be selected for removal

`--limit <COUNT>`
 Remove at most this many ordered candidates

`--dry-run`
 Preview the exact ordered removals without changing paths or registrations

`--yes`
 Confirm and apply cleanup removal; required unless `--dry-run` is supplied

`--json`
 Emit the complete stable machine-readable JSON payload

`-h, --help`
 Print help (see a summary with `'-h'`)

ait worktree prune-stale

Binary description: Select registrations whose path is missing or whose worktree runtime layout is detached. Pruning removes stale registration and alias state without deleting surviving checkout content. `--dry-run` previews the exact rows; applied pruning requires `--yes`.

```
Select registrations whose path is missing or whose worktree runtime layout is detached. Pruning
removes stale registration and alias state without deleting surviving checkout content.
--dry-run previews the exact rows; applied pruning requires --yes.
```

```
Usage: ait worktree prune-stale [OPTIONS]
```

```
Options:
```

```
--dry-run
```

```
    Preview missing and detached registrations without pruning them
```

```
--yes
```

```
    Confirm and apply stale-registration pruning; required unless --dry-run is supplied
```

```
--json
```

```
    Emit the complete stable machine-readable JSON payload
```

```
-h, --help
```

```
    Print help (see a summary with '-h')
```

ait worktree list

Binary description: List registered paths, current Lines, cached workspace status, cleanup classification, and current-context identity. `--refresh` verifies live content and may refresh derived runtime layout and status caches.

```
List registered paths, current Lines, cached workspace status, cleanup classification, and
current-context identity. --refresh verifies live content and may refresh derived runtime
layout and status caches.
```

```
Usage: ait worktree list [OPTIONS]
```

```
Options:
```

```
--json
```

```
    Emit the complete stable machine-readable JSON array
```

```
--refresh
```

```
    Verify live content and refresh derived runtime status for every worktree
```

```
-h, --help
```

```
    Print help (see a summary with '-h')
```

ait worktree sync

Binary description: Restore a complete registered worktree to a Line head and update its checked-out Snapshot, current Line, and runtime metadata. `NAME` selects one worktree; `--all` selects every available worktree and uses each one's current Line. `--all` cannot be combined with `NAME` or `--line`. Unsaved changes require `--force`, and `--dry-run` previews without applying the restore.

Restore a complete registered worktree to a Line head and update its checked-out Snapshot, current Line, and runtime metadata. NAME selects one worktree; --all selects every available worktree and uses each one's current Line. --all cannot be combined with NAME or --line. Unsaved changes require --force, and --dry-run previews without applying the restore.

Usage: ait worktree sync [OPTIONS] [NAME]

Arguments:

[NAME]

Registered worktree name; omit it to use the current runtime worktree binding

Options:

--all

Synchronize every live worktree to its own current Line; cannot be combined with NAME or --line

--line <LINE_NAME>

Synchronize one worktree to this Line head and make that Line current

--force

Overwrite unsaved changes while synchronizing

--dry-run

Preview complete restore plans without synchronizing content or metadata

--json

Emit the complete stable machine-readable JSON payload

-h, --help

Print help (see a summary with '-h')

ait worktree recreate

Binary description: Recreate the recorded path and alias of a Task-bound worktree whose registered path is missing. Recovery selects the first locally available current-Line head, fork Snapshot, or selected remote Patchset revision. An existing or unbound worktree is rejected; --dry-run validates and reports the candidate without creating its files.

Recreate the recorded path and alias of a Task-bound worktree whose registered path is missing. Recovery selects the first locally available current-Line head, fork Snapshot, or selected remote Patchset revision. An existing or unbound worktree is rejected; --dry-run validates and reports the candidate without creating its files.

Usage: ait worktree recreate [OPTIONS] [NAME]

Arguments:

[NAME]

Missing registered Task worktree name; omit it to use the current runtime binding

Options:

--dry-run

Validate recovery candidates, path, and alias without recreating the worktree

--json

Emit the complete stable machine-readable JSON payload

-h, --help

Print help (see a summary with '-h')

ait worktree recover-task

Binary description: Run from the main repository root to validate an active or draft remote Task and its draft or review Change, then recreate their local feature Line and Task-bound worktree. This command does not create a remote Task or Change. If a registration already exists but its path is missing, use worktree recreate instead.

Run from the main repository root to validate an active or draft remote Task and its draft or review Change, then recreate their local feature Line and Task-bound worktree. This command does not create a remote Task or Change. If a registration already exists but its path is missing, use `worktree recreate` instead.

Usage: `ait worktree recover-task [OPTIONS] --change <CHANGE> <TASK_ID>`

Arguments:

`<TASK_ID>`
Existing remote Task ID to recover

Options:

`--change <CHANGE>`
Existing remote Change ID or Task-owned Change reference

`--remote <REMOTE>`
Configured remote name; omit it to use the effective default remote

`--dry-run`
Validate remote identity, local Snapshots, and placement without creating a worktree

`--json`
Emit the complete stable machine-readable JSON payload

`-h, --help`
Print help (see a summary with `'-h'`)

ait worktree restore-owned-head

Binary description: Inspect first-parent history after the registered fork and retain the last contiguous Snapshot owned by the bound Task, Change, and worktree. Any first foreign Snapshot and its descendants are dropped from the Line head. The worktree must be Task-bound, clean, and outside a conflicted rebase; `--dry-run` reports the ownership decision without changing content or the head.

Inspect first-parent history after the registered fork and retain the last contiguous Snapshot owned by the bound Task, Change, and worktree. Any first foreign Snapshot and its descendants are dropped from the Line head. The worktree must be Task-bound, clean, and outside a conflicted rebase; `--dry-run` reports the ownership decision without changing content or the head.

Usage: `ait worktree restore-owned-head [OPTIONS] [NAME]`

Arguments:

`[NAME]`
Task-bound worktree name; omit it to use the current runtime worktree binding

Options:

`--dry-run`
Report the retained owned head and dropped foreign Snapshots without restoring

`--json`
Emit the complete stable machine-readable JSON payload

`-h, --help`
Print help (see a summary with `'-h'`)

ait worktree rebase

Binary description: Rebasing the current feature Line from its registered fork onto `--onto` or the recorded target base Line. Starting an applied rebase requires a clean worktree. `--dry-run` previews ancestry, writes, removals, and conflicts. A conflicted rebase must later use exactly one of `--continue` or `--abort`; neither continuation mode accepts `--dry-run`.

Rebase the current feature Line from its registered fork onto --onto or the recorded target base Line. Starting an applied rebase requires a clean worktree. --dry-run previews ancestry, writes, removals, and conflicts. A conflicted rebase must later use exactly one of --continue or --abort; neither continuation mode accepts --dry-run.

Usage: ait worktree rebase [OPTIONS] [NAME]

Arguments:

[NAME]
Registered worktree name; omit it to use the current runtime worktree binding

Options:

--onto <LINE_NAME>
Target base Line for a new rebase; omit it to use recorded target metadata

--continue
Snapshot the resolved conflicted workspace and complete its rebase

--abort
Discard conflicted workspace resolution and restore the original head

--dry-run
Preview a new rebase plan; cannot be combined with --continue or --abort

--json
Emit the complete stable machine-readable JSON payload

-h, --help
Print help (see a summary with '-h')

ait worktree remove

Binary description: Remove one or more named worktree registrations, managed aliases, and runtime identity files, leaving ordinary checkout content unless --delete-path is supplied. --all-stale performs the same stale-registration pruning as prune-stale and cannot be combined with names, --delete-path, or --force. --dry-run previews the exact plan; every applied removal requires --yes.

Remove one or more named worktree registrations, managed aliases, and runtime identity files, leaving ordinary checkout content unless --delete-path is supplied. --all-stale performs the same stale-registration pruning as prune-stale and cannot be combined with names, --delete-path, or --force. --dry-run previews the exact plan; every applied removal requires --yes.

Usage: ait worktree remove [OPTIONS] [NAME]...

Arguments:

[NAME]...
One or more registered worktree names; mutually exclusive with --all-stale

Options:

--all-stale
Prune every missing or detached registration; cannot be combined with names, --delete-path, or --force

--delete-path
Also delete each physical worktree path and managed Task Cargo build cache

--force
Permit explicit removal of dirty worktrees; does not bypass other safety checks

--dry-run
Preview exact removals without changing paths, aliases, or registrations

--yes
Confirm and apply worktree removal; required unless --dry-run is supplied

--json
Emit the complete stable machine-readable JSON payload

-h, --help
Print help (see a summary with '-h')

CLI · COMPLETE 1.1.1 REFERENCE

ait workflow - Complete Reference

Classify bounded edits, reconcile workflow state, prepare remote readiness, and drive configured closeout decisions.

AUDIENCE Coding agents and operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/workflow/>

Operating boundary

Purpose: Classify bounded edits, reconcile workflow state, prepare remote readiness, and drive configured closeout decisions.

Effect class: Guide is read-only; apply forms of reconciliation, readiness, and closeout mutate the stated scope.

Coverage: 5 public help surfaces under `ait workflow`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait workflow
ait workflow guide
ait workflow reconcile
ait workflow ready
ait workflow finish
```

Exact emitted help

ait workflow

Binary description: Prepare Changes for review, finish ready work, and repair workflow state.

```
Prepare Changes for review, finish ready work, and repair workflow state.

Usage: ait workflow <COMMAND>

Commands:
  guide      Show step-by-step guidance for common inventory and finish operations.
  reconcile  Inspect Task, Change, Line, worktree, closeout, and Plan bindings for repair;
             dry-run is the default and never changes Plan state.
  ready      Check or prepare one Change for review and remote finish; changes require --apply.
  finish     Check or apply remote Review and Policy requirements for one Change's selected
             Patchset, then safely finish the ready Change.
  help       Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

ait workflow guide

Binary description: Show step-by-step guidance for common inventory and finish operations.

```
Show step-by-step guidance for common inventory and finish operations.

Usage: ait workflow guide [TOPIC]

Arguments:
  [TOPIC]

Options:
  -h, --help  Print help
```

ait workflow reconcile

Binary description: Inspect Task, Change, Line, worktree, closeout, and Plan bindings for repair; dry-run is the default and never changes Plan state.

```
Inspect Task, Change, Line, worktree, closeout, and Plan bindings for repair; dry-run is the
default and never changes Plan state.

Usage: ait workflow reconcile [OPTIONS]

Options:
  --remote <REMOTE>
  --task <TASK>
  --dry-run
  --apply
  --safe-only
  --scheduled          Run one bounded remote-worker reconciliation pass; implies --safe-only
                        and requires a selected or default remote
  --limit <LIMIT>      [default: 100]
  --json
  -h, --help           Print help
```

ait workflow ready

Binary description: Check or prepare one Change for review and remote finish; changes require --apply.

```
Check or prepare one Change for review and remote finish; changes require --apply.

Usage: ait workflow ready [OPTIONS] <CHANGE_ID>

Arguments:
  <CHANGE_ID>

Options:
  --apply
  --snapshot-message <SNAPSHOT_MESSAGE>
  --summary <SUMMARY>
  --tests <TESTS>
  --lint <LINT>
  --security <SECURITY>
  --license <LICENSE>
  --author-mode <AUTHOR_MODE>
      [possible values: human_only, human_with_ai_assist, ai_with_human_review,
      ai_only_experimental]
  --model <MODEL>
  --remote <REMOTE>
  -h, --help           Print help
```

ait workflow finish

Binary description: Check or apply remote Review and Policy requirements for one Change's selected Patchset, then safely finish the ready Change.

Check or apply remote Review and Policy requirements for one Change's selected Patchset, then safely finish the ready Change.

Usage: ait workflow finish [OPTIONS] <CHANGE_ID>

Arguments:

<CHANGE_ID> Remote Change id to inspect or finish through reviewer-owned closeout.

Options:

- apply
Apply the safe next reviewer, policy, and atomic closeout actions instead of only showing state.
- review-message <REVIEW_MESSAGE>
Structured exact-Patchset AI review summary authored by the executing reviewer app.
- remote <REMOTE>
Use the named remote instead of the configured default remote.
- h, --help
Print help

CLI · COMPLETE 1.1.1 REFERENCE

ait line - Complete Reference

Inspect and manage Lines, switch or restore workspace state, merge recorded history, and clean eligible Line records.

AUDIENCE Developers and coding agents

<https://ait-native.dev/technical/v1.1.1/cli/reference/line/>

Operating boundary

Purpose: Inspect and manage Lines, switch or restore workspace state, merge recorded history, and clean eligible Line records.

Effect class: Mixed read and local mutation surface; switch, restore, merge, create, rename, archive, and cleanup alter local state.

Coverage: 10 public help surfaces under `ait line`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait line
ait line list
ait line create
ait line switch
ait line show
ait line archive
ait line rename
ait line delete
ait line merge
ait line cleanup
```

Exact emitted help

ait line

Binary description: Manage named Lines and their Snapshot heads. `ait branch` is a Git-friendly alias with the same subcommands, safety checks, and output; the recorded objects remain AIT Lines and Snapshots.

Manage named Lines and their Snapshot heads. `ait branch` is a Git-friendly alias with the same subcommands, safety checks, and output; the recorded objects remain AIT Lines and Snapshots.`

Usage: `ait line <COMMAND>`

Commands:

<code>list</code>	List active local Lines or Lines from a remote.
<code>create</code>	Create a new local Line at a Snapshot head.
<code>switch</code>	Select a local Line, optionally restoring its files.
<code>show</code>	Show one local or remote Line and its current head.
<code>archive</code>	Archive a local or remote Line without deleting its Snapshots.
<code>rename</code>	Rename a line while preserving its stable identity and reconciling bound pointers.
<code>delete</code>	Delete only a line ref after binding and unique-history protection checks.
<code>merge</code>	Merge one Line into the current Line with resumable conflict state and a two-parent Snapshot.
<code>cleanup</code>	Preview idle temporary Lines, or archive eligible candidates with <code>--yes</code> .
<code>help</code>	Print this message or the help of the given subcommand(s)

Options:

<code>-h, --help</code>	Print help (see a summary with <code>'-h'</code>)
-------------------------	--

ait line list

Binary description: List active local Lines or Lines from a remote.

```
List active local Lines or Lines from a remote.
```

```
Usage: ait line list [OPTIONS]
```

```
Options:
```

```
--all
--archived
--remote <REMOTE>
--json
-h, --help          Print help
```

ait line create

Binary description: Create a new local Line at a Snapshot head.

```
Create a new local Line at a Snapshot head.
```

```
Usage: ait line create [OPTIONS] <NAME>
```

```
Arguments:
```

```
<NAME>
```

```
Options:
```

```
--from-snapshot <FROM_SNAPSHOT> Start the new Line at this Snapshot instead of the
current Line head.
--switch                               Select the new Line without changing workspace files.
--json
-h, --help          Print help
```

ait line switch

Binary description: Select a local Line, optionally restoring its files.

```
Select a local Line, optionally restoring its files.
```

```
Usage: ait line switch [OPTIONS] <NAME>
```

```
Arguments:
```

```
<NAME>
```

```
Options:
```

```
--restore Restore the selected Line head's files into the workspace.
--force    Allow --restore to overwrite conflicting workspace changes.
--json
-h, --help Print help
```

ait line show

Binary description: Show one local or remote Line and its current head.

```
Show one local or remote Line and its current head.
```

```
Usage: ait line show [OPTIONS] [NAME]
```

```
Arguments:
```

```
[NAME]
```

```
Options:
```

```
--json
-h, --help Print help
```

ait line archive

Binary description: Archive a local or remote Line without deleting its Snapshots.

Archive a local or remote Line without deleting its Snapshots.

Usage: ait line archive [OPTIONS] <NAME>

Arguments:
<NAME>

Options:
--remote <REMOTE>
--json
-h, --help Print help

ait line rename

Binary description: Rename a line while preserving its stable identity and reconciling bound pointers.

Rename a line while preserving its stable identity and reconciling bound pointers.

Usage: ait line rename [OPTIONS] <OLD> <NEW>

Arguments:
<OLD>
<NEW>

Options:
--remote <REMOTE>
--json
-h, --help Print help

ait line delete

Binary description: Delete only a line ref after binding and unique-history protection checks.

Delete only a line ref after binding and unique-history protection checks.

Usage: ait line delete [OPTIONS] --yes <NAME>

Arguments:
<NAME>

Options:
--remote <REMOTE>
--yes
--json
-h, --help Print help

ait line merge

Binary description: Merge one Line into the current Line with resumable conflict state and a two-parent Snapshot.

Merge one Line into the current Line with resumable conflict state and a two-parent Snapshot.

Usage: ait line merge [OPTIONS] [SOURCE]

Arguments:
[SOURCE]

Options:
--message <MESSAGE>
--continue
--abort
--json
-h, --help Print help

ait line cleanup

Binary description: Preview idle temporary Lines, or archive eligible candidates with --yes.

Preview idle temporary Lines, or archive eligible candidates with --yes.

Usage: ait line cleanup [OPTIONS]

Options:

--idle-for <IDLE_FOR>	Minimum time since the Line's last activity, such as 7d, 12h, or 30m. [default: 7d]
--kind <CLEANUP_KIND>	Restrict cleanup to review_base, review, or wip Lines.
--limit <LIMIT>	Select at most this positive number of oldest candidates.
--include-protected	Include non-candidates and their protection reasons in the result.
--all	Show every selected row; protected rows still require
--include-protected.	
--yes	Archive the eligible candidates; omission is always a read-only
preview.	
--json	
-h, --help	Print help

CLI · COMPLETE 1.1.1 REFERENCE

ait git - Complete Reference

Import, export, and mirror history between Git and AIT's native repository model.

AUDIENCE Repository owners and migration operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/git/>

Operating boundary

Purpose: Import, export, and mirror history between Git and AIT's native repository model.

Effect class: History conversion and synchronization; direction, target, and mapping options determine the authority changed.

Coverage: 4 public help surfaces under `ait git`.

The blocks below preserve the 1.1.1 binary output in full, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait git
ait git import
ait git export
ait git mirror
```

Exact emitted help

`ait git`

Binary description: Import, export, or mirror history between Git and AIT.

```
Import, export, or mirror history between Git and AIT.

Usage: ait git <COMMAND>

Commands:
  import  Import Git commits, branches, and tags into AIT with a resumable immutable identity
         map.
  export  Export AIT lines, tags, and ordered Snapshot DAG history to Git without force-updating
         refs.
  mirror  Reconcile a Git endpoint and AIT through checkpointed, divergence-safe ref-set
         transactions.
  help    Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

`ait git import`

Binary description: Import Git commits, branches, and tags into AIT with a resumable immutable identity map.

```
Import Git commits, branches, and tags into AIT with a resumable immutable identity map.

Usage: ait git import [OPTIONS] <SOURCE>

Arguments:
  <SOURCE>  Git source to import; accepts a local repository path or a Git remote URL.

Options:
  --all-branches-and-tags  Import every Git branch and tag instead of only the source HEAD
                           branch.
  --dry-run                Inspect and validate the immutable import plan without writing
                           AIT data.
  --json                  Emit the complete machine-readable JSON result.
  -h, --help              Print help
```

ait git export

Binary description: Export AIT lines, tags, and ordered Snapshot DAG history to Git without force-updating refs.

```
Export AIT lines, tags, and ordered Snapshot DAG history to Git without force-updating refs.

Usage: ait git export [OPTIONS] <TARGET>

Arguments:
  <TARGET>  Local Git repository path to update, or a new path whose parent already exists.

Options:
  --all-lines-and-tags  Export every headed AIT Line and every AIT Tag instead of only the
                           current Line.
  --dry-run            Inspect and validate the immutable export plan without writing the
                           target or AIT data.
  --json              Emit the complete machine-readable JSON result.
  -h, --help          Print help
```

ait git mirror

Binary description: Reconcile a Git endpoint and AIT through checkpointed, divergence-safe ref-set transactions.

```
Reconcile a Git endpoint and AIT through checkpointed, divergence-safe ref-set transactions.

Usage: ait git mirror [OPTIONS] --direction <DIRECTION> <ENDPOINT>

Arguments:
  <ENDPOINT>  Git source for inbound mode; local Git path for outbound or bidirectional mode.

Options:
  --direction <DIRECTION>  Select the permitted change direction; one-way modes block
                           opposite-side changes and all modes block divergence. [possible values: inbound,
                           outbound, bidirectional]
  --dry-run                Classify and validate the complete branch and tag set without
                           writing changes.
  --json                  Emit the complete machine-readable JSON result.
  -h, --help              Print help
```

CLI · COMPLETE 1.1.1 COMPONENT REFERENCES

ait-agent - Complete Reference

Configure, inspect, and supervise optional Telegram, LINE, Discord, and Slack transport workers.

AUDIENCE Developers, integration maintainers, and operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/ait-agent/>

Operating boundary

Purpose: Configure, inspect, and supervise optional Telegram, LINE, Discord, and Slack transport workers.

Coverage: 43 public help surfaces emitted by `ait-agent`.

This page preserves the admitted 1.1.1 executable output in full. It is generated from the checksum-bound release binary, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait-agent
ait-agent telegram
ait-agent telegram add
ait-agent telegram list
ait-agent telegram status
ait-agent telegram start
ait-agent telegram stop
ait-agent telegram restart
ait-agent telegram logs
ait-agent telegram remove
ait-agent telegram supervisor
ait-agent telegram supervisor status
ait-agent telegram supervisor start
ait-agent telegram supervisor run
ait-agent telegram supervisor stop
ait-agent telegram supervisor restart
ait-agent line
ait-agent line add
ait-agent line list
ait-agent line status
ait-agent line start
ait-agent line stop
ait-agent line restart
ait-agent line logs
ait-agent line remove
ait-agent discord
ait-agent discord add
ait-agent discord list
ait-agent discord status
ait-agent discord start
ait-agent discord stop
ait-agent discord restart
ait-agent discord logs
ait-agent discord remove
ait-agent slack
ait-agent slack add
ait-agent slack list
ait-agent slack status
ait-agent slack start
ait-agent slack stop
ait-agent slack restart
ait-agent slack logs
ait-agent slack remove
```

Exact emitted help

ait-agent

```
Run optional ait external runtime workers.

Usage: ait-agent <COMMAND>

Commands:
  telegram  Run and configure Telegram agent workers.
  line      Run and configure LINE agent workers.
  discord   Run and configure Discord agent workers.
  slack     Run and configure Slack agent workers.
  help      Print this message or the help of the given subcommand(s)

Options:
  -h, --help      Print help
  -V, --version    Print version
```

ait-agent telegram

```
Run and configure Telegram agent workers.

Usage: ait-agent telegram [OPTIONS] [COMMAND]

Commands:
  add      Add or update a named Telegram worker config.
  list     List named Telegram worker configs.
  status   Show named Telegram worker runtime status.
  start    Start a named Telegram worker in daemon mode.
  stop     Stop a named Telegram worker.
  restart  Stop and restart a named Telegram worker in daemon mode.
  logs     Read the last N lines from a named Telegram worker's log.
  remove   Remove a named Telegram worker config.
  supervisor Supervise configured Telegram workers.
  help     Print this message or the help of the given subcommand(s)

Options:
  --mode <MODE>      [default: polling] [possible values: polling, webhook]
  --worker <WORKER>  [default: main]
  -h, --help          Print help
```

ait-agent telegram add

```
Add or update a named Telegram worker config.

Usage: ait-agent telegram add [OPTIONS] --token <TOKEN> <NAME>

Arguments:
  <NAME>

Options:
  --token <TOKEN>
  --username <USERNAME>
  --sync-state-path <SYNC_STATE_PATH>
  --pid-file <PID_FILE>
  --log-file <LOG_FILE>
  --env-path <ENV_PATH>
  --json
  -h, --help          Print help
```

ait-agent telegram list

```
List named Telegram worker configs.

Usage: ait-agent telegram list [OPTIONS]

Options:
  --json
  -h, --help  Print help
```

ait-agent telegram status

```
Show named Telegram worker runtime status.

Usage: ait-agent telegram status [OPTIONS] [NAME]

Arguments:
  [NAME]

Options:
  --json
  -h, --help  Print help
```

ait-agent telegram start

```
Start a named Telegram worker in daemon mode.

Usage: ait-agent telegram start [OPTIONS] <NAME>

Arguments:
  <NAME>

Options:
  --json
  -h, --help  Print help
```

ait-agent telegram stop

```
Stop a named Telegram worker.

Usage: ait-agent telegram stop [OPTIONS] <NAME>

Arguments:
  <NAME>

Options:
  --json
  -h, --help  Print help
```

ait-agent telegram restart

```
Stop and restart a named Telegram worker in daemon mode.

Usage: ait-agent telegram restart [OPTIONS] <NAME>

Arguments:
  <NAME>

Options:
  --json
  -h, --help  Print help
```

ait-agent telegram logs

```
Read the last N lines from a named Telegram worker's log.

Usage: ait-agent telegram logs [OPTIONS] <NAME>

Arguments:
  <NAME>

Options:
  --lines <LINES> [default: 100]
  --json
  -h, --help          Print help
```

ait-agent telegram remove

```
Remove a named Telegram worker config.

Usage: ait-agent telegram remove [OPTIONS] <NAME>

Arguments:
  <NAME>

Options:
  --json
  -h, --help  Print help
```

ait-agent telegram supervisor

```
Supervise configured Telegram workers.

Usage: ait-agent telegram supervisor <COMMAND>

Commands:
  status  Show runtime status for all Telegram workers.
  start   Start all configured Telegram workers.
  run     Continuously start stopped Telegram workers at an interval.
  stop    Stop all running Telegram workers.
  restart Restart all Telegram workers.
  help    Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

ait-agent telegram supervisor status

```
Show runtime status for all Telegram workers.

Usage: ait-agent telegram supervisor status [OPTIONS]

Options:
  --json
  -h, --help  Print help
```

ait-agent telegram supervisor start

```
Start all configured Telegram workers.

Usage: ait-agent telegram supervisor start [OPTIONS]

Options:
  --json
  -h, --help  Print help
```

ait-agent telegram supervisor run

```
Continuously start stopped Telegram workers at an interval.

Usage: ait-agent telegram supervisor run [OPTIONS]

Options:
  --interval-seconds <INTERVAL_SECONDS> [default: 30]
  --once
  --json
  -h, --help  Print help
```

ait-agent telegram supervisor stop

```
Stop all running Telegram workers.

Usage: ait-agent telegram supervisor stop [OPTIONS]

Options:
  --json
  -h, --help  Print help
```

ait-agent telegram supervisor restart

```
Restart all Telegram workers.

Usage: ait-agent telegram supervisor restart [OPTIONS]

Options:
  --json
  -h, --help  Print help
```

ait-agent line

```
Run and configure LINE agent workers.

Usage: ait-agent line <COMMAND>

Commands:
  add      Add or update a named LINE worker config.
  list
  status
  start
  stop
  restart
  logs
  remove
  help     Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

ait-agent line add

```
Add or update a named LINE worker config.

Usage: ait-agent line add [OPTIONS] --token <TOKEN> --secret <SECRET> <NAME>

Arguments:
  <NAME>

Options:
  --token <TOKEN>
  --secret <SECRET>
  --sync-state-path <SYNC_STATE_PATH>
  --pid-file <PID_FILE>
  --log-file <LOG_FILE>
  --env-path <ENV_PATH>
  --json
  -h, --help          Print help
```

ait-agent line list

```
Usage: ait-agent line list [OPTIONS]

Options:
  --json
  -h, --help  Print help
```

ait-agent line status

```
Usage: ait-agent line status [OPTIONS] [NAME]

Arguments:
  [NAME]

Options:
  --json
  -h, --help  Print help
```

ait-agent line start

```
Usage: ait-agent line start [OPTIONS] <NAME>
```

```
Arguments:  
  <NAME>
```

```
Options:  
  --json  
  -h, --help  Print help
```

ait-agent line stop

```
Usage: ait-agent line stop [OPTIONS] <NAME>
```

```
Arguments:  
  <NAME>
```

```
Options:  
  --json  
  -h, --help  Print help
```

ait-agent line restart

```
Usage: ait-agent line restart [OPTIONS] <NAME>
```

```
Arguments:  
  <NAME>
```

```
Options:  
  --json  
  -h, --help  Print help
```

ait-agent line logs

```
Usage: ait-agent line logs [OPTIONS] <NAME>
```

```
Arguments:  
  <NAME>
```

```
Options:  
  --lines <LINES>  [default: 100]  
  --json  
  -h, --help        Print help
```

ait-agent line remove

```
Usage: ait-agent line remove [OPTIONS] <NAME>
```

```
Arguments:  
  <NAME>
```

```
Options:  
  --json  
  -h, --help  Print help
```

ait-agent discord

```
Run and configure Discord agent workers.

Usage: ait-agent discord <COMMAND>

Commands:
  add      Add or update a named Discord worker config.
  list
  status
  start
  stop
  restart
  logs
  remove
  help     Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

ait-agent discord add

```
Add or update a named Discord worker config.

Usage: ait-agent discord add [OPTIONS] --application-id <APPLICATION_ID> --bot-token <BOT_TOKEN>
      <NAME>

Arguments:
  <NAME>

Options:
  --application-id <APPLICATION_ID>
  --bot-token <BOT_TOKEN>
  --sync-state-path <SYNC_STATE_PATH>
  --pid-file <PID_FILE>
  --log-file <LOG_FILE>
  --env-path <ENV_PATH>
  --json
  -h, --help                Print help
```

ait-agent discord list

```
Usage: ait-agent discord list [OPTIONS]

Options:
  --json
  -h, --help  Print help
```

ait-agent discord status

```
Usage: ait-agent discord status [OPTIONS] [NAME]

Arguments:
  [NAME]

Options:
  --json
  -h, --help  Print help
```

ait-agent discord start

```
Usage: ait-agent discord start [OPTIONS] <NAME>

Arguments:
  <NAME>

Options:
  --json
  -h, --help  Print help
```

ait-agent discord stop

```
Usage: ait-agent discord stop [OPTIONS] <NAME>

Arguments:
  <NAME>

Options:
  --json
  -h, --help  Print help
```

ait-agent discord restart

```
Usage: ait-agent discord restart [OPTIONS] <NAME>

Arguments:
  <NAME>

Options:
  --json
  -h, --help  Print help
```

ait-agent discord logs

```
Usage: ait-agent discord logs [OPTIONS] <NAME>

Arguments:
  <NAME>

Options:
  --lines <LINES> [default: 100]
  --json
  -h, --help          Print help
```

ait-agent discord remove

```
Usage: ait-agent discord remove [OPTIONS] <NAME>

Arguments:
  <NAME>

Options:
  --json
  -h, --help  Print help
```

ait-agent slack

```
Run and configure Slack agent workers.

Usage: ait-agent slack <COMMAND>

Commands:
  add      Add or update a named Slack worker config.
  list
  status
  start
  stop
  restart
  logs
  remove
  help     Print this message or the help of the given subcommand(s)

Options:
  -h, --help  Print help
```

ait-agent slack add

```
Add or update a named Slack worker config.

Usage: ait-agent slack add [OPTIONS] --app-token <APP_TOKEN> <NAME>

Arguments:
  <NAME>

Options:
  --app-token <APP_TOKEN>
  --sync-state-path <SYNC_STATE_PATH>
  --pid-file <PID_FILE>
  --log-file <LOG_FILE>
  --env-path <ENV_PATH>
  --json
  -h, --help                Print help
```

ait-agent slack list

```
Usage: ait-agent slack list [OPTIONS]

Options:
  --json
  -h, --help  Print help
```

ait-agent slack status

```
Usage: ait-agent slack status [OPTIONS] [NAME]

Arguments:
  [NAME]

Options:
  --json
  -h, --help  Print help
```

ait-agent slack start

```
Usage: ait-agent slack start [OPTIONS] <NAME>

Arguments:
  <NAME>

Options:
  --json
  -h, --help  Print help
```

ait-agent slack stop

```
Usage: ait-agent slack stop [OPTIONS] <NAME>

Arguments:
  <NAME>

Options:
  --json
  -h, --help  Print help
```

ait-agent slack restart

```
Usage: ait-agent slack restart [OPTIONS] <NAME>

Arguments:
  <NAME>

Options:
  --json
  -h, --help  Print help
```

ait-agent slack logs

```
Usage: ait-agent slack logs [OPTIONS] <NAME>

Arguments:
  <NAME>

Options:
  --lines <LINES>  [default: 100]
  --json
  -h, --help        Print help
```

ait-agent slack remove

```
Usage: ait-agent slack remove [OPTIONS] <NAME>

Arguments:
  <NAME>

Options:
  --json
  -h, --help  Print help
```

Capture authority

- Product version: 1.1.1
- Binary version: ait-agent 1.1.1
- Binary SHA-256: d0eb40a320e320bbefefd4faa11bdcfa0bac3fe1bff4cc3b1752de26983dc6d6
- Captured: 2026-09-02

CLI · COMPLETE 1.1.1 COMPONENT REFERENCES

ait-agent-worker - Complete Reference

Run one admitted native transport worker or one bounded signed-request or reply-provider transaction.

AUDIENCE Developers, integration maintainers, and operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/ait-agent-worker/>

Operating boundary

Purpose: Run one admitted native transport worker or one bounded signed-request or reply-provider transaction.

Coverage: 6 public help surfaces emitted by `ait-agent-worker`.

This page preserves the admitted 1.1.1 executable output in full. It is generated from the checksum-bound release binary, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait-agent-worker
ait-agent-worker capabilities
ait-agent-worker run
ait-agent-worker slack-command
ait-agent-worker discord-interaction
ait-agent-worker reply-provider
```

Exact emitted help

ait-agent-worker

```
Run Python-free AIT transport workers

Usage: ait-agent-worker <COMMAND>

Commands:
  capabilities      Report transport runners compiled into this executable
  run              Run one configured transport worker
  slack-command     Execute one signed Slack slash-command request from stdin
  discord-interaction Execute one signed Discord interaction request from stdin
  reply-provider    Generate one native Codex reply from a versioned provider request on
                    stdin
  help             Print this message or the help of the given subcommand(s)

Options:
  -h, --help      Print help
  -V, --version    Print version
```

ait-agent-worker capabilities

```
Report transport runners compiled into this executable

Usage: ait-agent-worker capabilities [OPTIONS]

Options:
  --json  Emit the stable machine-readable capability contract
  -h, --help  Print help
```

ait-agent-worker run

Run one configured transport worker

Usage: ait-agent-worker run [OPTIONS] --transport <TRANSPORT> --worker <WORKER>
--event-loop-backend <EVENT_LOOP_BACKEND> --shard <SHARD>

Options:

--transport <TRANSPORT>
Transport kind: telegram, discord, slack, or line
--worker <WORKER>
Named worker from .ait/agent-workers.json
--event-loop-backend <EVENT_LOOP_BACKEND>
Rust event-loop backend selected by launch planning
--shard <SHARD>
Zero-based Rust reactor shard assignment
--console-mode <CONSOLE_MODE>
Run the long-lived service or one stdin Telegram webhook transaction [default: service] [possible values: service, webhook]
-h, --help
Print help

ait-agent-worker slack-command

Execute one signed Slack slash-command request from stdin

Usage: ait-agent-worker slack-command [OPTIONS] --signature <SIGNATURE> --signature-timestamp <SIGNATURE_TIMESTAMP>

Options:

--worker <WORKER>
Named Slack worker from .ait/agent-workers.json [default: main]
--signature <SIGNATURE>
Slack request signature paired with the exact stdin payload
--signature-timestamp <SIGNATURE_TIMESTAMP>
Slack request Unix timestamp used by signature verification
-h, --help
Print help

ait-agent-worker discord-interaction

Execute one signed Discord interaction request from stdin

Usage: ait-agent-worker discord-interaction [OPTIONS] --signature <SIGNATURE>
--signature-timestamp <SIGNATURE_TIMESTAMP>

Options:

--worker <WORKER>
Named Discord worker from .ait/agent-workers.json [default: main]
--signature <SIGNATURE>
Discord Ed25519 signature paired with the exact stdin payload
--signature-timestamp <SIGNATURE_TIMESTAMP>
Discord interaction timestamp used by signature verification
-h, --help
Print help

ait-agent-worker reply-provider

Generate one native Codex reply from a versioned provider request on stdin

Usage: ait-agent-worker reply-provider

Options:

-h, --help Print help

Capture authority

- Product version: 1.1.1
- Binary version: ait-agent-worker 1.1.1
- Binary SHA-256: 23997a12b6237523517cf1729ebd745d17a8cc23dab7610b6d3faf1587d534d8

- Captured: 2026-09-02

CLI · COMPLETE 1.1.1 COMPONENT REFERENCES

ait-server - Complete Reference

Initialize, probe, or run durable remote Repository authority and Worker Job coordination.

AUDIENCE Developers, integration maintainers, and operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/ait-server/>

Operating boundary

Purpose: Initialize, probe, or run durable remote Repository authority and Worker Job coordination.

Coverage: 1 public help surface emitted by `ait-server`.

This page preserves the admitted 1.1.1 executable output in full. It is generated from the checksum-bound release binary, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

ait-server

Exact emitted help

ait-server

```
Usage:
  ait-server [--data <absolute-path>] [--listen <ip:port>] [--init-if-missing]
    [--defer-ci-admission]
  ait-server init [--data <absolute-path>]
  ait-server probe [--data <absolute-path>] [--defer-ci-admission]
  ait-server --startup-probe
  ait-server --help
  ait-server --version
```

With no arguments, `ait-server` starts with the platform user-data root, initializes it on first use, listens on 127.0.0.1:8088, and defers the startup CI RAM-workspace probe. An explicit `--data` or `AIT_NATIVE_SERVER_DATA` root does not enable guarded initialization unless `--init-if-missing` is supplied. `--listen` is accepted only when starting the server. `--defer-ci-admission` skips only the startup RAM-workspace probe; managed CI allocation still fails closed until a memory-backed root is configured.

Capture authority

- Product version: 1.1.1
- Binary version: `ait-server 1.1.1`
- Binary SHA-256: `fe460c47c0c0bbcf4be217776a120232c0df9f9c0e5e7cac2ed6aa21d98d31b8`
- Captured: 2026-09-02

CLI · COMPLETE 1.1.1 COMPONENT REFERENCES

ait-runner - Complete Reference

Diagnose, execute, claim, or serve typed server-owned CI work.

AUDIENCE Developers, integration maintainers, and operators

<https://ait-native.dev/technical/v1.1.1/cli/reference/ait-runner/>

Operating boundary

Purpose: Diagnose, execute, claim, or serve typed server-owned CI work.

Coverage: 5 public help surfaces emitted by `ait-runner`.

This page preserves the admitted 1.1.1 executable output in full. It is generated from the checksum-bound release binary, including every emitted argument, option, default, possible value, conflict, and requirement.

Complete command tree

```
ait-runner
ait-runner doctor
ait-runner execute
ait-runner run-job
ait-runner serve
```

Exact emitted help

ait-runner

```
Native execution plane for explicit ait-server CI jobs

Usage: ait-runner <COMMAND>

Commands:
  doctor  Verify the live ait-server health contract without claiming a job
  execute Execute one typed native request from stdin or a named file
  run-job Claim, execute, and finish one explicit Binary Worker Job pair
  serve   Poll for compatible native jobs, maintain their lease, and deliver results
  help    Print this message or the help of the given subcommand(s)

Options:
  -h, --help      Print help
  -V, --version   Print version
```

ait-runner doctor

```
Verify the live ait-server health contract without claiming a job

Usage: ait-runner doctor --server <SERVER>

Options:
  --server <SERVER>
  -h, --help      Print help
```

ait-runner execute

```
Execute one typed native request from stdin or a named file

Usage: ait-runner execute [OPTIONS]

Options:
  --request <REQUEST>      [default: -]
  --source-root <SOURCE_ROOT> [default: .]
  --attempt-root <ATTEMPT_ROOT>
  -h, --help              Print help
```

ait-runner run-job

Claim, execute, and finish one explicit Binary Worker Job pair

Usage: ait-runner run-job [OPTIONS] --server <SERVER> --repository-index <REPOSITORY_INDEX>
--worker-job-index <WORKER_JOB_INDEX>

Options:

--server <SERVER>
--repository-index <REPOSITORY_INDEX>
--worker-job-index <WORKER_JOB_INDEX>
--source-root <SOURCE_ROOT> [default: .]
--attempt-root <ATTEMPT_ROOT>
-h, --help Print help

ait-runner serve

Poll for compatible native jobs, maintain their lease, and deliver results

Usage: ait-runner serve [OPTIONS] --server <SERVER> --worker-id <WORKER_ID>

Options:

--server <SERVER>
--worker-id <WORKER_ID>
--repository-index <REPOSITORY_INDEXES>
--once
--poll-interval-ms <POLL_INTERVAL_MS> [default: 1000]
--heartbeat-interval-ms <HEARTBEAT_INTERVAL_MS> [default: 30000]
--source-root <SOURCE_ROOT> [default: .]
--attempt-root <ATTEMPT_ROOT>
-h, --help Print help

Capture authority

- Product version: 1.1.1
- Binary version: ait-runner 1.1.1
- Binary SHA-256: 14c326d8a1a83f15a3aca3d39bd57aa49f5eabafc5740ad19cbcbd114663ba94
- Captured: 2026-09-02

VERSION AUTHORITY

1.1.1 Source Authority

This PDF is a navigable public reference, not a separate product contract. Use the exact 1.1.1 source, Release, distribution contract, and CLI parser below when release authority matters.

PUBLIC SOURCE	https://github.com/weita2026/ait-native/tree/v1.1.1
PUBLIC COMMIT	d248b29d3d988867366287d91e57b6c5bace716b
RELEASE	https://github.com/weita2026/ait-native/releases/tag/v1.1.1
DISTRIBUTION SHA-256	f2290e405520454e7926f09cdda6a1fe4be1b60300f03d4222918bedc9dd3efa
CLI PARSER SHA-256	829ccf69699c28bcca872cd2b191d256ed8aef4b4e8ada0590fb1e0aach38215
AIT-CORE	SNP-ED7593DBF982
AIT-SERVER	SNP-0CCD7DD2A077
AIT-RUNNER	SNP-35C9C133D2EE
AIT-PYTHON	SNP-756C731A4CC0
AIT-NODE	SNP-55E90D0A81F1

For the current online edition, search, and route-specific updates, open ait-native.dev/technical/.