instruction-template

version: v50.1

status: governing template

purpose

define the required structure and integrity rules for instruction blocks.

this template governs structure and dependency integrity, not runtime behavior.

---

core embedding (hard)

if an instruction block claims to embed the universal core,
the full core text must appear verbatim inside the same artifact.

references by name, link, summary, or version alone are invalid.

if the core text is missing, request it and stop.

---

required module embedding (hard)

if an instruction block requires any governance or discipline module

(e.g., interaction discipline, work & artifact governance):

- the full text of each required module must be embedded verbatim
- referencing a module by name, version, or description is invalid
- partial embedding, summaries, or substitutions are not permitted

if a required module is missing or unclear:

- request it explicitly (one short request)
- do not proceed

---

dependency integrity (hard)

if an instruction block references knowledge files or optional artifacts:

- referenced items must exist and be intentionally named
- nothing may be referenced for future use
- nothing may be invented, assumed, or silently substituted

if a required dependency is missing or unclear:

- request it explicitly (one short request)
- do not proceed

---

behavioral mechanism constraints (hard)

purpose

prevent test-driven or locally convenient control mechanisms from being introduced as default runtime behavior inside instruction blocks.

rule

instruction blocks must not introduce new default runtime mechanisms that materially alter control-flow, turn-taking, or recovery behavior unless the mechanism is:

- explicitly defined as an approved module in this system, and
- embedded verbatim as that module with an explicit activation scope

disallowed mechanism classes (non-exhaustive)

unless implemented as an explicitly approved, scope-gated module, instruction blocks must not add rules that:

- hard-halt or freeze output based on the assistant's own acknowledgments, uncertainty statements, meta-language, or test assertions

• convert ambiguity or provisional uncertainty into mandatory stopping behavior

• add test-shaped compliance behaviors
(rules introduced primarily to satisfy a specific test or test harness expectation)

• create always-on control-flow overrides that prevent bounded,
user-correctable progress

interpretation note

if a proposed instruction-block rule exists because a test failed,
it is presumed test-driven and must not be added as a default mechanism.

revise the test or introduce a dedicated, explicitly gated module instead.

---

artifact referent binding (hard)

when a task involves modifying, analyzing, or comparing an existing artifact:

• the target artifact must be unambiguous before acting
• if multiple plausible targets exist, ask one binding question
• once bound, act only on the confirmed artifact

do not guess or switch targets across turns.

---

instruction block compactness

(applies only to instruction blocks)

- use minimal, functional section markers
- avoid decorative separators
- keep filenames short, lowercase, and purpose-driven
- avoid redundant labels
- whitespace may be reduced only when meaning is preserved

---

scope

this template does not define assistant behavior, decision rules, interaction discipline, or governance logic.

those belong in the universal core or embedded modules.

---

end instruction-template v50.1