

Package: wf (via r-universe)

September 18, 2026

Title Artificial Intelligence Workflow Tools

Version 0.1.0

Description Manage skills for large language model coding agents.
Supports installing skills from 'GitHub' or local directories,
tracking versions in a lock file, and keeping installations
current. Installations can be scoped to a single project or
shared globally across projects.

URL <https://christophertkenny.com/wf/>,
<https://github.com/christopherkenny/wf>

BugReports <https://github.com/christopherkenny/wf/issues>

Depends R (>= 4.1.0)

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports cli (>= 3.6.0), fs, gh, jsonlite, rlang (>= 1.1.0), yaml

Suggests testthat (>= 3.0.0), withr

Config/testthat/edition 3

Config/Needs/website christopherkenny/ctktemplate

Config/pak/sysreqs cmake git make libuv1-dev libssl-dev

Repository <https://christopherkenny.r-universe.dev>

Date/Publication 2026-05-20 19:52:48 UTC

RemoteUrl <https://github.com/christopherkenny/wf>

RemoteRef HEAD

RemoteSha 9299ca43ad3be4f31ac5973fa907b93104300785

Contents

add_agent	2
add_hook	4
add_rule	5
add_skill	6
agent_path	7
check_agents	8
check_hooks	9
check_rules	10
check_skills	11
find_agent	12
find_hook	12
find_rule	13
find_skill	14
hook_path	15
init_agent	15
init_hook	16
init_rule	17
init_skill	18
list_agents	18
list_hooks	19
list_rules	20
list_skills	21
register_hook	22
remove_agent	23
remove_hook	24
remove_rule	25
remove_skill	26
rule_path	27
settings_path	27
skill_path	28
update_agents	29
update_hooks	30
update_rules	31
update_skills	31

Index	33
--------------	-----------

add_agent	<i>Install an agent</i>
-----------	-------------------------

Description

Installs a custom agent from a GitHub repository or a local file into an agents directory. Agents are single Markdown files with YAML frontmatter specifying at minimum a name and description.

Usage

```
add_agent(source, agent = NULL, path = NULL, overwrite = FALSE)
```

Arguments

source	One of: • A GitHub URL pointing to a repo, e.g. "https://github.com/owner/repo". • A GitHub URL pointing to a subdirectory or file, e.g. "https://github.com/owner/repo/tree/main". • A GitHub shorthand, e.g. "owner/repo". • A local file path pointing to a Markdown file.
agent	The agent to install. One of: • A bare agent name (without .md), e.g. "code-reviewer". The agent is read from agents/<agent>.md within the repository. • A path to the agent file, e.g. "r-lib/mirai/AGENT.md" or "r-lib/mirai/". Used directly as the path within the repository. Ignored when source already points to a specific path via /tree/... or /blob/....
path	The agents directory. Can be one of: • A known coding agent name such as 'claude_code', 'cursor', or 'github_copilot' (see agent_path() for the full list) to use that agent's conventional project-scope path. • A character string giving the directory path directly. • NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviro (e.g. with usethis::edit_r_environ()) to avoid the prompt.
overwrite	If FALSE (the default), an error is raised if the agent is already installed. Set to TRUE to replace it.

Value

The path to the installed agent file, invisibly.

Examples

```
src <- tempfile(fileext = '.md')
writeLines(
  c('---', 'name: example', 'description: An example agent.', '---'),
  src
)
add_agent(src, path = tempfile())
```

add_hook	<i>Install a hook script</i>
----------	------------------------------

Description

Installs a hook script from a GitHub repository or a local file into a hooks directory, then registers it in the corresponding settings.json so it runs on the specified lifecycle event.

Usage

```
add_hook(  
  source,  
  event,  
  hook = NULL,  
  matcher = NULL,  
  path = NULL,  
  settings = NULL,  
  overwrite = FALSE,  
  timeout = NULL,  
  async = FALSE  
)
```

Arguments

source	One of: • A GitHub URL pointing to a repo, e.g. "https://github.com/owner/repo". • A GitHub shorthand, e.g. "owner/repo". • A local file path pointing to an executable script (.sh, .R, .py, etc.).
event	The lifecycle event to attach the hook to. One of "PreToolUse", "PostToolUse", "UserPromptSubmit", "Stop", or "SubagentStop".
hook	For multi-hook repositories that store scripts under a hooks/ subdirectory, the name of the hook to install (without the file extension), e.g. hook = "lint-staged". When supplied, the script is read from hooks/<hook>.<ext> within the repository. Ignored when source already points to a specific file.
matcher	An optional tool-name pattern (for "PreToolUse" and "PostToolUse" events) used to filter which tool calls trigger the hook, e.g. "Bash Edit". When NULL (the default), the hook applies to all tool calls for the event.
path	The hooks directory. Can be one of: • A known coding agent name such as 'claude_code', 'cursor', or 'github_copilot' (see hook_path() for the full list) to use that agent's conventional project-scope path. • A character string giving the directory path directly. • NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renvron (e.g. with usethis::edit_r_envron()) to avoid the prompt.

settings	Path to the settings.json file where the hook will be registered. When NULL (the default), defaults to settings.json in the parent directory of path (e.g. if path is .claude/hooks, uses .claude/settings.json).
overwrite	If FALSE (the default), an error is raised if the hook is already installed. Set to TRUE to replace it.
timeout	Optional timeout in seconds for the hook command.
async	If TRUE, the hook runs asynchronously. Default is FALSE.

Value

The path to the installed hook script, invisibly.

Examples

```
tmp_hook <- tempfile(fileext = '.sh')
writeLines(c('#!/bin/bash', 'echo hello'), tmp_hook)
tmp_dir <- tempfile()
tmp_settings <- tempfile(fileext = '.json')
add_hook(tmp_hook,
  event = 'PreToolUse', path = tmp_dir,
  settings = tmp_settings
)
```

add_rule	<i>Install a rule</i>
----------	-----------------------

Description

Installs a rule from a GitHub repository or a local file into a rules directory. Rules are Markdown files with optional YAML frontmatter. The rule name comes from the frontmatter if present, otherwise from the filename stem.

Usage

```
add_rule(source, rule = NULL, path = NULL, overwrite = FALSE)
```

Arguments

source	One of: • A GitHub URL pointing to a repo, e.g. "https://github.com/owner/repo". • A GitHub URL pointing to a subdirectory or file, e.g. "https://github.com/owner/repo/tree/main". • A GitHub shorthand, e.g. "owner/repo". • A local file path pointing to a Markdown file.
rule	The rule to install. One of: • A bare rule name (without .md), e.g. "testing". The rule is read from rules/<rule>.md within the repository.

	• A path to the rule file, e.g. "r-lib/testing/testing.md" or "r-lib/testing/". Used directly as the path within the repository. Ignored when source already points to a specific path via /tree/... or /blob/....
path	The rules directory. Can be one of: • A known coding agent name such as 'claude_code', 'cursor', or 'github_copilot' (see <code>rule_path()</code> for the full list) to use that agent's conventional project-scope path. • A character string giving the directory path directly. • NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviron (e.g. with <code>usethis::edit_r_environ()</code>) to avoid the prompt.
overwrite	If FALSE (the default), an error is raised if the rule is already installed. Set to TRUE to replace it.

Value

The path to the installed rule file, invisibly.

Examples

```
src <- tempfile(fileext = '.md')
writeLines(
  c('---', 'name: example', 'description: An example rule.', '---'),
  src
)
add_rule(src, path = tempfile())
```

add_skill	<i>Install a skill</i>
-----------	------------------------

Description

Installs a skill from a GitHub repository or a local directory into a skills directory.

Usage

```
add_skill(source, skill = NULL, path = NULL, overwrite = FALSE)
```

Arguments

source	One of: • A GitHub URL pointing to a repo, e.g. "https://github.com/owner/repo". • A GitHub URL pointing to a subdirectory, e.g. "https://github.com/owner/repo/tree/main/pa". • A GitHub shorthand, e.g. "owner/repo". • A local directory path containing a SKILL.md file.
--------	---

skill	The skill to install. One of: • A bare skill name, e.g. "proofread". The skill is read from skills/<skill> within the repository. • A path to the skill directory, e.g. "r-lib/mirai" or "r-lib/mirai/". Used directly as the subdirectory path. • A path to the SKILL.md file, e.g. "r-lib/mirai/SKILL.md". The filename is stripped and the containing directory is used. Ignored when source already points to a specific subdirectory via /tree/... or /blob/....
path	The skills directory. Can be one of: • A known coding agent name such as 'claude_code', 'cursor', or 'github_copilot' (see skill_path() for the full list) to use that agent's conventional project-scope path. • A character string giving the directory path directly. • NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renvron (e.g. with usethis::edit_r_envron()) to avoid the prompt.
overwrite	If FALSE (the default), an error is raised if the skill is already installed. Set to TRUE to replace it.

Value

The path to the installed skill directory, invisibly.

Examples

```
src <- tempfile()
dir.create(src)
writeLines(
  c('---', 'name: example', 'description: An example skill.', '---'),
  file.path(src, 'SKILL.md')
)
add_skill(src, path = tempfile())
```

agent_path	<i>Get the conventional agent path for a coding agent</i>
------------	---

Description

Returns the conventional directory path where custom agents for a given coding agent are stored. The path is not expanded (i.e., ~ is not resolved to the home directory). Use [fs::path_expand\(\)](#) if you need an absolute path.

Usage

```
agent_path(agent = NULL, scope = c("project", "global"))
```

Arguments

agent	One of "claude_code" (or its alias "claude"), "openclaw", "codex", "cursor", "gemini_cli", "github_copilot" (or its alias "copilot"), or "posit_ai" (or its alias "posit"). If NULL (the default), the agent is resolved in order: (1) the WF_AGENT environment variable, (2) a scan of the current working directory for a recognised agent config folder (.claude, .cursor, etc.), and (3) a final fallback to "claude_code". Set WF_AGENT in your .Renviron (e.g. with usethis::edit_r_environ()) to avoid passing agent every time.
scope	One of "project" (a path relative to the current working directory, suitable for committing to version control) or "global" (a path in the user's home directory, available across all projects).

Value

A length-1 character vector giving the conventional agents path.

Examples

```
agent_path('claude_code', 'project')
agent_path('claude', 'project') # alias for claude_code
agent_path('cursor', 'global')
agent_path() # auto-detects from WF_AGENT, dir scan, or falls back to claude_code
```

check_agents	<i>Check installed agents for available updates</i>
--------------	---

Description

Compares each installed agent's recorded commit SHA against the latest commit on GitHub. Local agents are reported as not updatable.

Usage

```
check_agents(path = NULL)
```

Arguments

path	The agents directory. Can be one of: • A known coding agent name such as 'claude_code', 'cursor', or 'github_copilot' (see agent_path() for the full list) to use that agent's conventional project-scope path. • A character string giving the directory path directly. • NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviron (e.g. with usethis::edit_r_environ()) to avoid the prompt.
------	--

Value

A data frame with columns:

- name: agent name.
- installed_sha: the SHA recorded at install time (NA for local).
- latest_sha: the current HEAD SHA on GitHub (NA for local or on network failure).
- update_available: TRUE if installed and latest SHAs differ.

Examples

```
check_agents(tempfile())
```

check_hooks

Check installed hooks for available updates

Description

Compares each installed hook's recorded commit SHA against the latest commit on GitHub. Local hooks are reported as not updatable.

Usage

```
check_hooks(path = NULL)
```

Arguments

- | | |
|------|---|
| path | <p>The hooks directory. Can be one of:</p> <ul style="list-style-type: none"> • A known coding agent name such as 'claude_code', 'cursor', or 'github_copilot' (see hook_path() for the full list) to use that agent's conventional project-scope path. • A character string giving the directory path directly. • NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviro (e.g. with usethis::edit_r_enviro()) to avoid the prompt. |
|------|---|

Value

A data frame with columns:

- name: hook name.
- installed_sha: the SHA recorded at install time (NA for local).
- latest_sha: the current HEAD SHA on GitHub (NA for local or on network failure).
- update_available: TRUE if installed and latest SHAs differ.

Examples

```
check_hooks(tempfile())
```

check_rules	<i>Check installed rules for available updates</i>
-------------	--

Description

Compares each installed rule's recorded commit SHA against the latest commit on GitHub. Local rules are reported as not updatable.

Usage

```
check_rules(path = NULL)
```

Arguments

path	The rules directory. Can be one of: • A known coding agent name such as 'claude_code', 'cursor', or 'github_copilot' (see rule_path() for the full list) to use that agent's conventional project-scope path. • A character string giving the directory path directly. • NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviro (e.g. with usethis::edit_r_enviro()) to avoid the prompt.
------	---

Value

A data frame with columns:

- name: rule name.
- installed_sha: the SHA recorded at install time (NA for local).
- latest_sha: the current HEAD SHA on GitHub (NA for local or on network failure).
- update_available: TRUE if installed and latest SHAs differ.

Examples

```
check_rules(tempfile())
```

check_skills	<i>Check installed skills for available updates</i>
--------------	---

Description

Compares each installed skill's recorded commit SHA against the latest commit on GitHub. Local skills are reported as not updatable.

Usage

```
check_skills(path = NULL)
```

Arguments

path	The skills directory. Can be one of: • A known coding agent name such as 'claude_code', 'cursor', or 'github_copilot' (see skill_path() for the full list) to use that agent's conventional project-scope path. • A character string giving the directory path directly. • NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviro (e.g. with usethis::edit_r_enviro()) to avoid the prompt.
------	--

Value

A data frame with columns:

- name: skill name.
- installed_sha: the SHA recorded at install time (NA for local).
- latest_sha: the current HEAD SHA on GitHub (NA for local or on network failure).
- update_available: TRUE if installed and latest SHAs differ.

Examples

```
check_skills(tempfile())
```

find_agent	<i>Search for agents on GitHub</i>
------------	------------------------------------

Description

Searches GitHub for repositories tagged with an agent topic and matching a keyword query. Searches across all supported agent topic conventions (e.g. claude-agent, cursor-agent).

Usage

```
find_agent(query)
```

Arguments

query Keyword to search for.

Value

A data frame with columns:

- name: repository name.
- description: repository description.
- owner: repository owner login.
- url: full URL of the repository.
- stars: number of GitHub stars.

Examples

```
## Not run:  
# Requires GitHub API access; may fail due to rate limiting  
find_agent('code-review')  
  
## End(Not run)
```

find_hook	<i>Search for hooks on GitHub</i>
-----------	-----------------------------------

Description

Searches GitHub for repositories tagged with a hook topic and matching a keyword query. Searches across all supported hook topic conventions (e.g. claude-hook, cursor-hook).

Usage

```
find_hook(query)
```

Arguments

query Keyword to search for.

Value

A data frame with columns:

- name: repository name.
- description: repository description.
- owner: repository owner login.
- url: full URL of the repository.
- stars: number of GitHub stars.

Examples

```
## Not run:  
# Requires GitHub API access; may fail due to rate limiting  
find_hook('linting')  
  
## End(Not run)
```

find_rule	<i>Search for rules on GitHub</i>
-----------	-----------------------------------

Description

Searches GitHub for repositories tagged with a rule topic and matching a keyword query. Searches across all supported rule topic conventions (e.g. `claude-rule`, `cursor-rule`).

Usage

```
find_rule(query)
```

Arguments

query Keyword to search for.

Value

A data frame with columns:

- name: repository name.
- description: repository description.
- owner: repository owner login.
- url: full URL of the repository.
- stars: number of GitHub stars.

Examples

```
## Not run:  
# Requires GitHub API access; may fail due to rate limiting  
find_rule('testing')  
  
## End(Not run)
```

find_skill	<i>Search for skills on GitHub</i>
------------	------------------------------------

Description

Searches GitHub for repositories tagged with a skill topic and matching a keyword query. Searches across all supported agent topic conventions (e.g. `claude-skill`, `cursor-skill`).

Usage

```
find_skill(query)
```

Arguments

query	Keyword to search for.
-------	------------------------

Value

- A data frame with columns:
- name: repository name.
 - description: repository description.
 - owner: repository owner login.
 - url: full URL of the repository.
 - stars: number of GitHub stars.

Examples

```
## Not run:  
# Requires GitHub API access; may fail due to rate limiting  
find_skill('rstats')  
  
## End(Not run)
```

hook_path	<i>Get the path to a coding agent's hooks directory</i>
-----------	---

Description

Returns the conventional directory path where hook scripts for a given coding agent are stored. The path is not expanded (i.e., ~ is not resolved to the home directory). Use `fs::path_expand()` if you need an absolute path.

Usage

```
hook_path(agent = NULL, scope = c("project", "global"))
```

Arguments

agent	One of "claude_code" (or its alias "claude"), "openclaw", "codex", "cursor", "gemini_cli", "github_copilot" (or its alias "copilot"), or "posit_ai" (or its alias "posit"). If NULL (the default), the agent is resolved in order: (1) the WF_AGENT environment variable, (2) a scan of the current working directory for a recognised agent config folder (.claude, .cursor, etc.), and (3) a final fallback to "claude_code". Set WF_AGENT in your .Renviro (e.g. with <code>usethis::edit_r_environ()</code>) to avoid passing agent every time.
scope	One of "project" (a path relative to the current working directory, suitable for committing to version control) or "global" (a path in the user's home directory, available across all projects).

Value

A length-1 character vector giving the conventional hooks directory path.

Examples

```
hook_path('claude_code', 'project')
hook_path('claude', 'global') # alias for claude_code
hook_path('cursor', 'project')
hook_path() # auto-detects from WF_AGENT, dir scan, or falls back to claude_code
```

init_agent	<i>Create a new agent template</i>
------------	------------------------------------

Description

Creates a new agent file at path/name.md containing a template ready to be filled in.

Usage

```
init_agent(name, path = NULL)
```

Arguments

name	Agent name. Must be 1-64 characters, lowercase alphanumeric with single hyphens (no consecutive --), and cannot start or end with a hyphen. Consider using a gerund or role form (e.g. "code-reviewer", "test-writer").
path	Directory in which to create the agent file. Can be one of: • A known coding agent name such as "claude_code" or "github_copilot" to use that agent's conventional project-scope path (see agent_path() for the full list). • A character string giving the directory path directly. • NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviron (e.g. with usethis::edit_r_environ()) to avoid the prompt.

Value

The path to the new agent file, invisibly.

Examples

```
init_agent('my-agent', tempfile())
```

init_hook	Create a new hook template
-----------	----------------------------

Description

Creates a new shell script hook file at path/name.sh containing a minimal template ready to be filled in. After editing the script, register it with [register_hook\(\)](#) or install it with [add_hook\(\)](#).

Usage

```
init_hook(name, path = NULL)
```

Arguments

name	Hook name. Must be 1-64 characters, lowercase alphanumeric with single hyphens (no consecutive --), and cannot start or end with a hyphen. Consider using a kebab-case verb form (e.g. "lint-on-save", "check-format").
path	Directory in which to create the hook file. Can be one of: • A known coding agent name such as "claude_code" or "github_copilot" to use that agent's conventional project-scope path (see hook_path() for the full list). • A character string giving the directory path directly. • NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviron (e.g. with usethis::edit_r_environ()) to avoid the prompt.

Value

The path to the new hook file, invisibly.

Examples

```
init_hook('my-hook', tempfile())
```

init_rule	Create a new rule template
-----------	----------------------------

Description

Creates a new rule file at path/name.md containing a template ready to be filled in.

Usage

```
init_rule(name, path = NULL)
```

Arguments

name	Rule name. Must be 1-64 characters, lowercase alphanumeric with single hyphens (no consecutive --), and cannot start or end with a hyphen. Consider a descriptive noun form (e.g. "testing", "code-style").
path	Directory in which to create the rule file. Can be one of: • A known coding agent name such as "claude_code" or "github_copilot" to use that agent's conventional project-scope path (see rule_path() for the full list). • A character string giving the directory path directly. • NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviron (e.g. with usethis::edit_r_environ()) to avoid the prompt.

Value

The path to the new rule file, invisibly.

Examples

```
init_rule('my-rule', tempfile())
```

init_skill	Create a new skill template
------------	-----------------------------

Description

Creates a new skill directory at path/name/ containing a template SKILL.md file ready to be filled in.

Usage

```
init_skill(name, path = NULL)
```

Arguments

- | | |
|------|--|
| name | Skill name. Must be 1-64 characters, lowercase alphanumeric with single hyphens (no consecutive --), and cannot start or end with a hyphen. Consider using a gerund form (e.g. "parsing-logs"). |
| path | Directory in which to create the skill. The skill directory itself will be path/name. Can be one of: <ul style="list-style-type: none"> A known agent name such as "claude_code" or "github_copilot" to use that agent's conventional project-scope path (see skill_path() for the full list). A character string giving the directory path directly. NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviro (e.g. with usethis::edit_r_enviro()) to avoid the prompt. |

Value

The path to the new skill directory, invisibly.

Examples

```
init_skill('my-skill', tempfile())
```

list_agents	List installed agents
-------------	-----------------------

Description

Returns a data frame describing all agents installed in an agents directory.

Usage

```
list_agents(path = NULL)
```

Arguments

path	The agents directory. Can be one of: • A known coding agent name such as 'claude_code', 'cursor', or 'github_copilot' (see agent_path() for the full list) to use that agent's conventional project-scope path. • A character string giving the directory path directly. • NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviron (e.g. with usethis::edit_r_environ()) to avoid the prompt.
------	---

Value

- A data frame with columns:
- name: agent name from frontmatter (or filename stem).
 - description: agent description from frontmatter.
 - source: the source URL or local path the agent was installed from.
 - installed_at: ISO 8601 timestamp of when the agent was installed.

Examples

```
list_agents(tempfile())
```

list_hooks	<i>List hooks in a settings file</i>
------------	--------------------------------------

Description

Returns a data frame of all hooks configured in a coding agent's settings.json file.

Usage

```
list_hooks(  
  path = NULL,  
  settings = NULL,  
  agent = NULL,  
  scope = c("project", "local", "global")  
)
```

Arguments

path	The hooks directory. When supplied, the file column in the returned data frame will contain the path to the installed script file for hooks that were installed with <code>add_hook()</code> . Can be one of: • A known coding agent name to use that agent’s conventional hooks path. • A character string giving the directory path directly. • NULL (the default), in which case file will be NA for all rows.
settings	Path to the <code>settings.json</code> file to read. When NULL (the default), resolved from agent, scope, and the <code>WF_AGENT</code> environment variable.
agent, scope	Passed to <code>settings_path()</code> to locate the settings file when settings is NULL. Defaults resolve to the project-scope settings of the detected coding agent.

Value

- A data frame with columns:
- event: the lifecycle event name (e.g. "PreToolUse").
 - matcher: the tool-name pattern, or NA if none.
 - command: the shell command to execute.
 - file: path to the installed script file, or NA if not tracked.

Examples

```
tmp <- tempfile(fileext = '.json')
register_hook('PreToolUse', 'echo hello', path = tmp)
list_hooks(settings = tmp)
```

list_rules	<i>List installed rules</i>
------------	-----------------------------

Description

Returns a data frame describing all rules installed in a rules directory.

Usage

```
list_rules(path = NULL)
```

Arguments

path	The rules directory. Can be one of: • A known coding agent name such as 'claude_code', 'cursor', or 'github_copilot' (see <code>rule_path()</code> for the full list) to use that agent’s conventional project-scope path. • A character string giving the directory path directly.
------	---

- NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviron (e.g. with `usethis::edit_r_environ()`) to avoid the prompt.

Value

A data frame with columns:

- name: rule name from frontmatter (or filename stem).
- description: rule description from frontmatter.
- source: the source URL or local path the rule was installed from.
- installed_at: ISO 8601 timestamp of when the rule was installed.

Examples

```
list_rules(tempfile())
```

list_skills	<i>List installed skills</i>
-------------	------------------------------

Description

Returns a data frame describing all skills installed in a skills directory.

Usage

```
list_skills(path = NULL)
```

Arguments

- | | |
|------|--|
| path | The skills directory. Can be one of: <ul style="list-style-type: none">• A known coding agent name such as 'claude_code', 'cursor', or 'github_copilot' (see <code>skill_path()</code> for the full list) to use that agent's conventional project-scope path.• A character string giving the directory path directly.• NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviron (e.g. with <code>usethis::edit_r_environ()</code>) to avoid the prompt. |
|------|--|

Value

A data frame with columns:

- name: skill name from frontmatter (or filename stem).
- description: skill description from frontmatter.
- source: the source URL or local path the skill was installed from.
- installed_at: ISO 8601 timestamp of when the skill was installed.

Examples

```
list_skills(tempfile())
```

register_hook	Register a hook command in the settings file
---------------	--

Description

Adds a shell command hook to a coding agent’s settings.json file. Hooks run automatically on agent events such as "PreToolUse" or "UserPromptSubmit". To install a hook script from GitHub or a local file, use [add_hook\(\)](#) instead.

Usage

```
register_hook(  
    event,  
    command,  
    matcher = NULL,  
    timeout = NULL,  
    async = FALSE,  
    agent = NULL,  
    scope = c("project", "local", "global"),  
    path = NULL  
)
```

Arguments

event	The lifecycle event to attach the hook to. One of "PreToolUse", "PostToolUse", "UserPromptSubmit", "Stop", or "SubagentStop".
command	The shell command to execute when the hook fires.
matcher	An optional tool-name pattern (for "PreToolUse" and "PostToolUse" events) used to filter which tool calls trigger the hook, e.g. "Bash Edit". When NULL (the default), the hook applies to all tool calls for the event.
timeout	Optional timeout in seconds for the hook command.
async	If TRUE, the hook runs asynchronously and the agent does not wait for it to complete. Default is FALSE.
agent, scope, path	Passed to settings_path() to locate the settings file. Defaults resolve to the project-scope settings of the detected coding agent.

Value

The path to the modified settings file, invisibly.

Examples

```
tmp <- tempfile(fileext = '.json')
register_hook('PreToolUse', 'echo hello', path = tmp)
```

remove_agent	<i>Remove an installed agent</i>
--------------	----------------------------------

Description

Deletes an agent file from an agents directory and removes it from the lock file.

Usage

```
remove_agent(name, path = NULL, force = FALSE)
```

Arguments

name	The name of the agent to remove (without the .md extension).
path	The agents directory. Can be one of: • A known coding agent name such as 'claude_code', 'cursor', or 'github_copilot' (see agent_path() for the full list) to use that agent’s conventional project-scope path. • A character string giving the directory path directly. • NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviron (e.g. with usethis::edit_r_environ()) to avoid the prompt.
force	If FALSE (the default), prompts for confirmation in interactive sessions. Set to TRUE to skip the prompt.

Value

The name of the removed agent, invisibly.

Examples

```
src <- tempfile(fileext = '.md')
writeLines(
  c('---', 'name: example', 'description: An example agent.', '---'),
  src
)
tmp <- tempfile()
add_agent(src, path = tmp)
remove_agent('example', tmp, force = TRUE)
```

remove_hook	<i>Remove an installed hook</i>
-------------	---------------------------------

Description

Deletes a hook script from the hooks directory, removes its registration from `settings.json`, and removes it from the lock file.

Usage

```
remove_hook(name, path = NULL, settings = NULL, force = FALSE)
```

Arguments

name	The name of the hook to remove (the script filename stem, e.g. "lint-staged" for <code>lint-staged.sh</code>).
path	The hooks directory. Can be one of: - A known coding agent name such as 'claude_code', 'cursor', or 'github_copilot' (see hook_path() for the full list) to use that agent's conventional project-scope path. - A character string giving the directory path directly. - NULL (the default), in which case the path is resolved from the <code>WF_AGENT</code> environment variable, or by prompting in interactive sessions. Set <code>WF_AGENT</code> in your <code>.Renvron</code> (e.g. with <code>usethis::edit_r_envron()</code>) to avoid the prompt.
settings	Path to the <code>settings.json</code> file where the hook is registered. When NULL (the default), defaults to <code>settings.json</code> in the parent directory of path.
force	If FALSE (the default), prompts for confirmation in interactive sessions. Set to TRUE to skip the prompt.

Value

The name of the removed hook, invisibly.

Examples

```
tmp_hook <- tempfile(fileext = '.sh')
writeLines(c('#!/bin/bash', 'echo hello'), tmp_hook)
tmp_dir <- tempdir()
tmp_settings <- tempfile(fileext = '.json')
add_hook(tmp_hook,
  event = 'PreToolUse', path = tmp_dir,
  settings = tmp_settings
)
remove_hook(fs::path_ext_remove(basename(tmp_hook)), tmp_dir,
  settings = tmp_settings, force = TRUE
)
```

remove_rule	<i>Remove an installed rule</i>
-------------	---------------------------------

Description

Deletes a rule file from a rules directory and removes it from the lock file.

Usage

```
remove_rule(name, path = NULL, force = FALSE)
```

Arguments

name	The name of the rule to remove (without the .md extension).
path	The rules directory. Can be one of: • A known coding agent name such as 'claude_code', 'cursor', or 'github_copilot' (see rule_path() for the full list) to use that agent's conventional project-scope path. • A character string giving the directory path directly. • NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviro (e.g. with usethis::edit_r_environ()) to avoid the prompt.
force	If FALSE (the default), prompts for confirmation in interactive sessions. Set to TRUE to skip the prompt.

Value

The name of the removed rule, invisibly.

Examples

```
src <- tempfile(fileext = '.md')
writeLines(
  c('---', 'name: example', 'description: An example rule.', '---'),
  src
)
tmp <- tempfile()
add_rule(src, path = tmp)
remove_rule('example', tmp, force = TRUE)
```

remove_skill	<i>Remove an installed skill</i>
--------------	----------------------------------

Description

Deletes a skill directory from a skills directory and removes it from the lock file.

Usage

```
remove_skill(name, path = NULL, force = FALSE)
```

Arguments

name	The name of the skill to remove.
path	The skills directory. Can be one of: • A known coding agent name such as 'claude_code', 'cursor', or 'github_copilot' (see skill_path() for the full list) to use that agent's conventional project-scope path. • A character string giving the directory path directly. • NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviron (e.g. with usethis::edit_r_environ()) to avoid the prompt.
force	If FALSE (the default), prompts for confirmation in interactive sessions. Set to TRUE to skip the prompt.

Value

The name of the removed skill, invisibly.

Examples

```
src <- tempfile()
dir.create(src)
writeLines(
  c('---', 'name: example', 'description: An example skill.', '---'),
  file.path(src, 'SKILL.md')
)
tmp <- tempfile()
add_skill(src, path = tmp)
remove_skill('example', tmp, force = TRUE)
```

rule_path	<i>Get the conventional rule path for a coding agent</i>
-----------	--

Description

Returns the conventional directory path where rules for a given coding agent are stored. The path is not expanded (i.e., ~ is not resolved to the home directory). Use [fs::path_expand\(\)](#) if you need an absolute path.

Usage

```
rule_path(agent = NULL, scope = c("project", "global"))
```

Arguments

agent	One of "claude_code" (or its alias "claude"), "openclaw", "codex", "cursor", "gemini_cli", "github_copilot" (or its alias "copilot"), or "posit_ai" (or its alias "posit"). If NULL (the default), the agent is resolved in order: (1) the WF_AGENT environment variable, (2) a scan of the current working directory for a recognised agent config folder (.claude, .cursor, etc.), and (3) a final fallback to "claude_code". Set WF_AGENT in your .Renviron (e.g. with usethis::edit_r_environ()) to avoid passing agent every time.
scope	One of "project" (a path relative to the current working directory, suitable for committing to version control) or "global" (a path in the user's home directory, available across all projects).

Value

A length-1 character vector giving the conventional rules path.

Examples

```
rule_path('claude_code', 'project')
rule_path('claude', 'project') # alias for claude_code
rule_path('cursor', 'global')
rule_path() # auto-detects from WF_AGENT, dir scan, or falls back to claude_code
```

settings_path	<i>Get the path to a coding agent's settings file</i>
---------------	---

Description

Returns the path to the settings.json file where hooks are configured for a given coding agent and scope. The path is not expanded (i.e., ~ is not resolved to the home directory). Use [fs::path_expand\(\)](#) if you need an absolute path.

Usage

```
settings_path(agent = NULL, scope = c("project", "local", "global"))
```

Arguments

agent	One of "claude_code" (or its alias "claude"), "openclaw", "codex", "cursor", "gemini_cli", "github_copilot" (or its alias "copilot"), or "posit_ai" (or its alias "posit"). If NULL (the default), the agent is resolved in order: (1) the WF_AGENT environment variable, (2) a scan of the current working directory for a recognised agent config folder (.claude, .cursor, etc.), and (3) a final fallback to "claude_code". Set WF_AGENT in your .Renviro (e.g. with usethis::edit_r_environ()) to avoid passing agent every time.
scope	One of: • "project": {agent}/settings.json in the current directory. • "local": {agent}/settings.local.json (gitignored, for personal overrides). • "global": ~/.{agent}/settings.json.

Value

A length-1 character vector giving the path to the settings file.

Examples

```
settings_path('claude_code', 'project')
settings_path('claude', 'local') # alias for claude_code
settings_path('cursor', 'global')
```

skill_path	<i>Get the conventional skill path for an agent</i>
------------	---

Description

Returns the conventional directory path where skills for a given agent are stored. The path is not expanded (i.e., ~ is not resolved to the home directory). Use [fs::path_expand\(\)](#) if you need an absolute path.

Usage

```
skill_path(agent = NULL, scope = c("project", "global"))
```

Arguments

agent	One of "claude_code" (or its alias "claude"), "openclaw", "codex", "cursor", "gemini_cli", "github_copilot" (or its alias "copilot"), or "posit_ai" (or its alias "posit"). If NULL (the default), the agent is resolved in order: (1) the WF_AGENT environment variable, (2) a scan of the current working directory for a recognised agent config folder (.claude, .cursor, etc.), and (3) a final fallback to "claude_code". Set WF_AGENT in your .Renviron (e.g. with usethis::edit_r_environ()) to avoid passing agent every time.
scope	One of "project" (a path relative to the current working directory, suitable for committing to version control) or "global" (a path in the user's home directory, available across all projects).

Value

A length-1 character vector giving the conventional skill path.

Examples

```
skill_path('claude_code', 'project')
skill_path('claude', 'project') # alias for claude_code
skill_path('github_copilot', 'project')
skill_path('copilot', 'project') # alias for github_copilot
skill_path('posit_ai', 'project')
skill_path('posit', 'project') # alias for posit_ai
skill_path('cursor', 'global')
skill_path() # auto-detects from WF_AGENT, dir scan, or falls back to claude_code
```

update_agents	<i>Update installed agents</i>
---------------	--------------------------------

Description

Checks each installed agent for available updates and re-installs any that have a newer version on GitHub.

Usage

```
update_agents(path = NULL)
```

Arguments

path	The agents directory to update. Can be one of: • A known coding agent name such as "claude_code", "cursor", or "github_copilot" (see agent_path() for the full list) to use that agent's conventional project-scope path. • A character string giving the directory path directly.
------	--

- NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviron (e.g. with `usethis::edit_r_environ()`) to avoid the prompt.

Value

A character vector of updated agent names, invisibly.

Examples

```
update_agents(tempfile())
```

update_hooks	<i>Update installed hooks</i>
--------------	-------------------------------

Description

Checks each installed hook for available updates and re-installs any that have a newer version on GitHub.

Usage

```
update_hooks(path = NULL, settings = NULL)
```

Arguments

path	The hooks directory to update. Can be one of: • A known coding agent name such as "claude_code", "cursor", or "github_copilot" (see <code>hook_path()</code> for the full list) to use that agent's conventional project-scope path. • A character string giving the directory path directly. • NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviron (e.g. with <code>usethis::edit_r_environ()</code>) to avoid the prompt.
settings	Path to the settings.json file where hooks are registered. When NULL (the default), defaults to settings.json in the parent directory of path.

Value

A character vector of updated hook names, invisibly.

Examples

```
update_hooks(tempfile())
```

update_rules	<i>Update installed rules</i>
--------------	-------------------------------

Description

Checks each installed rule for available updates and re-installs any that have a newer version on GitHub.

Usage

```
update_rules(path = NULL)
```

Arguments

- | | |
|------|--|
| path | The rules directory to update. Can be one of: <ul style="list-style-type: none">• A known coding agent name such as "claude_code", "cursor", or "github_copilot" (see rule_path() for the full list) to use that agent's conventional project-scope path.• A character string giving the directory path directly.• NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviron (e.g. with usethis::edit_r_environ()) to avoid the prompt. |
|------|--|

Value

A character vector of updated rule names, invisibly.

Examples

```
update_rules(tempfile())
```

update_skills	<i>Update installed skills</i>
---------------	--------------------------------

Description

Checks each installed skill for available updates and re-installs any that have a newer version on GitHub.

Usage

```
update_skills(path = NULL)
```

Arguments

- path The skills directory to update. Can be one of:
- A known agent name such as "claude_code", "cursor", or "github_copilot" (see [skill_path\(\)](#) for the full list) to use that agent's conventional project-scope path.
 - A character string giving the directory path directly.
 - NULL (the default), in which case the path is resolved from the WF_AGENT environment variable, or by prompting in interactive sessions. Set WF_AGENT in your .Renviron (e.g. with [usethis::edit_r_environ\(\)](#)) to avoid the prompt.

Value

A character vector of updated skill names, invisibly.

Examples

```
update_skills(tempfile())
```

Index

add_agent, 2
add_hook, 4
add_hook(), 16, 20, 22
add_rule, 5
add_skill, 6
agent_path, 7
agent_path(), 3, 8, 16, 19, 23, 29

check_agents, 8
check_hooks, 9
check_rules, 10
check_skills, 11

find_agent, 12
find_hook, 12
find_rule, 13
find_skill, 14
fs::path_expand(), 7, 15, 27, 28

hook_path, 15
hook_path(), 4, 9, 16, 24, 30

init_agent, 15
init_hook, 16
init_rule, 17
init_skill, 18

list_agents, 18
list_hooks, 19
list_rules, 20
list_skills, 21

register_hook, 22
register_hook(), 16
remove_agent, 23
remove_hook, 24
remove_rule, 25
remove_skill, 26
rule_path, 27
rule_path(), 6, 10, 17, 20, 25, 31

settings_path, 27
settings_path(), 20, 22
skill_path, 28
skill_path(), 7, 11, 18, 21, 26, 32

update_agents, 29
update_hooks, 30
update_rules, 31
update_skills, 31
usethis::edit_r_environ(), 3, 4, 6–11,
15–19, 21, 23–32