

Package: rsprefs (via r-universe)

July 24, 2024

Title Manage and Sync Your RStudio Preferences
Version 0.0.2
Description Snapshot and restore different versions of your RStudio user preferences. Sync your user preferences via GitHub gist.
License MIT + file LICENSE
URL <https://github.com/gadenbuie/rsprefs>,
<http://pkg.garrickadenbuie.com/rsprefs>
BugReports <https://github.com/gadenbuie/rsprefs/issues>
Depends R (>= 3.5)
Imports checkmate, cli, fs, gh, jsonlite, purrr, rappdirs, rlang, rstudioapi
Config/Needs/website gadenbuie/grkgdown
Encoding UTF-8
LazyData true
Roxygen list(markdown = TRUE)
RoxygenNote 7.2.3
Repository <https://gadenbuie.r-universe.dev>
RemoteUrl <https://github.com/gadenbuie/rsprefs>
RemoteRef main
RemoteSha 41f863f16cb66803adeaca11a60baaa9bcc4bc17

Contents

prefs_reset_defaults	2
prefs_rstudio	3
prefs_rstudio_v	12
prefs_schema	12
snapshot_prefs	13

Index	15
--------------	-----------

prefs_reset_defaults *Reset Preferences to the RStudio Default*

Description

Returns currently set preferences to RStudio's original, built-in default.

Usage

```
prefs_reset_defaults(source = c("project", "user"), verbose = FALSE, ...)
```

Arguments

source	<p>The source of the current preference value. Preferences are set at different levels, from lowest to highest precedence:</p> <ul style="list-style-type: none"> • "default" are RStudio's built-in defaults • "computed" are detected or supplied from external sources, e.g. the path to <code>git</code> on your system • "system" are derived from system-wide <code>rstudio-prefs.json</code> • "user" are set by the user for all sessions (global options) • "project" are set by the current project session <p>The default is "user", since these are the settings you set yourself for all projects. You can include any number of sources, or "all" to include all preferences regardless of source.</p>
verbose	Prints or suppress informative output
...	<p>Arguments passed on to snapshot_prefs_save</p> <p>include Names of RStudio preferences to include. If provided, only these preferences are included. See prefs_rstudio for preference names.</p> <p>exclude Names of RStudio preferences to exclude from the snapshot. See prefs_rstudio for all of the preference names.</p> <p>exclude_os_prefs Excludes operating-system or machine-dependent system preferences from the snapshot or the snapshot restore.</p>

Value

Resets preferences to their built-in defaults, returning the current preferences invisibly as a list. You can also return to the preferences prior to `prefs_reset_defaults()` with [snapshot_prefs_undo\(\)](#).

Examples

```
## Not run:
prefs_reset_defaults()

## End(Not run)
```

```
prefs_rstudio      prefs_rstudio
```

Description

The RStudio User Preferences, their description and default value.

Usage

```
prefs_rstudio
```

Format

A list with 244 items:

```
run_rprofile_on_resume Whether to run .Rprofile again after resuming a suspended R session.
save_workspace Whether to save the workspace to an .Rdata file after the R session ends.
load_workspace Whether to load the workspace when the R session begins.
initial_working_directory The initial working directory for new R sessions.
cran_mirror The CRAN mirror to use.
bioconductor_mirror_name The name of the default Bioconductor mirror.
bioconductor_mirror_url The URL of the default Bioconductor mirror.
always_save_history Whether to always save the R console history.
remove_history_duplicates Whether to remove duplicate entries from the R console history.
show_last_dot_value Show the result of the last expression (.Last.value) in the Environment
pane.
line_ending_conversion The line ending format to use when saving files.
use_newlines_in_makefiles Whether to use newlines when saving Makefiles.
windows_terminal_shell The terminal shell to use on Windows.
posix_terminal_shell The terminal shell to use on POSIX operating systems (MacOS and Linux).
custom_shell_command The fully qualified path to the custom shell command to use in the Ter-
minal tab.
custom_shell_options The command-line options to pass to the custom shell command.
show_line_numbers Show line numbers in RStudio's code editor.
relative_line_numbers Show relative, rather than absolute, line numbers in RStudio's code ed-
itor.
highlight_selected_word Highlight the selected word in RStudio's code editor.
highlight_selected_line Highlight the selected line in RStudio's code editor.
panes Layout of panes in the RStudio workbench.
allow_source_columns Whether to enable the ability to add source columns to display.
```

`use_spaces_for_tab` Whether to insert spaces when pressing the Tab key.

`num_spaces_for_tab` The number of spaces to insert when pressing the Tab key.

`auto_detect_indentation` Whether to automatically detect indentation settings from file contents.

`show_margin` Whether to show the margin guide in the RStudio code editor.

`blinking_cursor` Whether to flash the cursor off and on.

`margin_column` The number of columns of text after which the margin is shown.

`show_invisibles` Whether to show invisible characters, such as spaces and tabs, in the RStudio code editor.

`indent_guides` Style for indentation guides in the RStudio code editor.

`continue_comments_on_newline` Whether to continue comments (by inserting the comment character) after adding a new line.

`highlight_web_link` Whether web links in comments are clickable.

`editor_keybindings` The keybindings to use in the RStudio code editor.

`insert_matching` Whether to insert matching pairs, such as `()` and `[]`, when the first is typed.

`insert_spaces_around_equals` Whether to insert spaces around the equals sign in R code.

`insert_parens_after_function_completion` Whether to insert parentheses after function completions.

`tab_multiline_completion` Whether to attempt completion of multiple-line statements when pressing Tab.

`tab_completion` Whether to attempt completion of statements when pressing Tab.

`show_help_tooltip_on_idle` Whether to show help tooltips for functions when the cursor has not been recently moved.

`surround_selection` Which kinds of delimiters can be used to surround the current selection.

`enable_snippets` Whether to enable code snippets in the RStudio code editor.

`code_completion` When to use auto-completion for R code in the RStudio code editor.

`code_completion_other` When to use auto-completion for other languages (such as JavaScript and SQL) in the RStudio code editor.

`console_code_completion` Whether to always use code completion in the R console.

`code_completion_delay` The number of milliseconds to wait before offering code suggestions.

`code_completion_characters` The number of characters in a symbol that can be entered before completions are offered.

`show_function_signature_tooltips` Whether to show function signature tooltips during auto-completion.

`show_diagnostics_r` Whether to show diagnostic messages (such as syntax and usage errors) for R code as you type.

`show_diagnostics_cpp` Whether to show diagnostic messages for C++ code as you type.

`show_diagnostics_yaml` Whether to show diagnostic messages for YAML code as you type.

`show_diagnostics_other` Whether to show diagnostic messages for other types of code (not R, C++, or YAML).

`style_diagnostics` Whether to show style diagnostics (suggestions for improving R code style)

`diagnostics_on_save` Whether to check code for problems after saving it.

`background_diagnostics` Whether to run code diagnostics in the background, as you type.

`background_diagnostics_delay_ms` The number of milliseconds to delay before running code diagnostics in the background.

`diagnostics_in_r_function_calls` Whether to run diagnostics in R function calls.

`check_arguments_to_r_function_calls` Whether to check arguments to R function calls.

`check_unexpected_assignment_in_function_call` Whether to check for unexpected variable assignments inside R function calls.

`warn_if_no_such_variable_in_scope` Whether to generate a warning if a variable is used without being defined in the current scope.

`warn_variable_defined_but_not_used` Whether to generate a warning if a variable is defined without being used in the current scope

`auto_discover_package_dependencies` Whether to automatically discover and offer to install missing R package dependencies.

`auto_append_newline` Whether to ensure that source files end with a newline character.

`strip_trailing_whitespace` Whether to strip trailing whitespace from each line when saving.

`restore_source_document_cursor_position` Whether to save the position of the cursor when a file is closed, restore it when the file is opened.

`reindent_on_paste` Whether to automatically re-indent code when it's pasted into RStudio.

`vertically_align_arguments_indent` Whether to vertically align arguments to R function calls during automatic indentation.

`soft_wrap_r_files` Whether to soft-wrap R source files, wrapping the text for display without inserting newline characters.

`soft_wrap_rmd_files` Whether to soft-wrap R Markdown files (and similar types such as R HTML and R Notebooks)

`focus_console_after_exec` Whether to focus the R console after executing an R command from a script.

`fold_style` The style of folding to use.

`save_before_sourcing` Whether to automatically save scripts before executing them.

`syntax_color_console` Whether to use syntax highlighting in the R console.

`highlight_console_errors` Whether to display error, warning, and message output in a different color.

`scroll_past_end_of_document` Whether to allow scrolling past the end of a file.

`highlight_r_function_calls` Whether to highlight R function calls in the code editor.

`color_preview` Whether to show preview for named and hexadecimal colors.

`rainbow_parentheses` Whether to highlight parentheses in a variety of colors.

`rainbow_fenced_divs` Whether to highlight fenced divs in a variety of colors.

`console_line_length_limit` The maximum number of characters to display in a single line in the R console.

`console_max_lines` The maximum number of console actions to store and display in the console scrollbar buffer.

`ansi_console_mode` How to treat ANSI escape codes in the console.

`limit_visible_console` Whether to only show a limited window of the total console output

`show_inline_toolbar_for_r_code_chunks` Whether to show a toolbar on code chunks in R Markdown documents.

`highlight_code_chunks` Whether to highlight code chunks in R Markdown documents with a different background color.

`save_files_before_build` Whether to save all open, unsaved files before building the project.

`font_size_points` The default editor font size, in points.

`help_font_size_points` The help panel font size, in points.

`editor_theme` The name of the color theme to apply to the text editor in RStudio.

`server_editor_font_enabled` Whether to use a custom editor font in RStudio Server.

`server_editor_font` The name of the fixed-width editor font to use with RStudio Server.

`default_encoding` The default character encoding to use when saving files.

`toolbar_visible` Whether to show the toolbar at the top of the RStudio workbench.

`default_project_location` The directory path under which to place new projects by default.

`source_with_echo` Whether to echo R code when sourcing it.

`default_sweave_engine` The default engine to use when processing Sweave documents.

`default_latex_program` The default program to use when processing LaTeX documents.

`use_roxygen` Whether to use Roxygen for documentation.

`use_dataimport` Whether to use RStudio's data import feature.

`pdf_previewer` The program to use to preview PDF files after generation.

`always_enable_rnw_concordance` Whether to always enable the concordance for RNW files.

`insert_numbered_latex_sections` Whether to insert numbered sections in LaTeX.

`spelling_dictionary_language` The language of the spelling dictionary to use for spell checking.

`spelling_custom_dictionaries` The list of custom dictionaries to use when spell checking.

`document_load_lint_delay` The number of milliseconds to wait before linting a document after it is loaded.

`ignore_uppercase_words` Whether to ignore words in uppercase when spell checking.

`ignore_words_with_numbers` Whether to ignore words with numbers in them when spell checking.

`real_time_spellchecking` Whether to enable real-time spellchecking by default.

`navigate_to_build_error` Whether to navigate to build errors.

`packages_pane_enabled` Whether to enable RStudio's Packages pane.

`cpp_template` C++ template.

`restore_source_documents` Whether to restore the last opened source documents when RStudio starts up.

`handle_errors_in_user_code_only` Whether to handle errors only when user code is on the stack.

`auto_expand_error_tracebacks` Whether to automatically expand tracebacks when an error occurs.

`check_for_updates` Whether to check for new versions of RStudio when RStudio starts.

`show_internal_functions` Whether to show functions without source references in the Traceback pane while debugging.

`shiny_viewer_type` Where to display Shiny applications when they are run.

`shiny_background_jobs` Whether to run Shiny applications as background jobs.

`plumber_viewer_type` Where to display Shiny applications when they are run.

`document_author` The default name to use as the document author when creating new documents.

`rmd_auto_date` Use current date when rendering document

`rmd_preferred_template_path` The path to the preferred R Markdown template.

`rmd_viewer_type` Where to display R Markdown documents when they have completed rendering.

`show_publish_diagnostics` Whether to show verbose diagnostic information when publishing content.

`enable_cloud_publish_ui` Whether to show UI for publishing content to Posit Cloud.

`publish_check_certificates` Whether to check remote server SSL certificates when publishing content.

`use_publish_ca_bundle` Whether to use a custom certificate authority (CA) bundle when publishing content.

`publish_ca_bundle` The path to the custom certificate authority (CA) bundle to use when publishing content.

`rmd_chunk_output_inline` Whether to show chunk output inline for ordinary R Markdown documents.

`show_doc_outline_rmd` Whether to show the document outline by default when opening R Markdown documents.

`auto_run_setup_chunk` Whether to automatically run an R Markdown document's Setup chunk before running other chunks.

`hide_console_on_chunk_execute` Whether to hide the R console when executing inline R Markdown chunks.

`execution_behavior` The unit of R code to execute when the Execute command is invoked.

`show_terminal_tab` Whether to show the Terminal tab.

`terminal_local_echo` Whether to use local echo in the Terminal.

`terminal_websockets` Whether to use websockets to communicate with the shell in the Terminal tab.

`terminal_close_behavior` Whether to close the terminal pane after the shell exits.

`terminal_track_environment` Whether to track and save changes to system environment variables in the Terminal.

`terminal_bell_style` Terminal bell style

`terminal_renderer` Terminal rendering engine: canvas is faster, dom may be needed for some browsers or graphics cards

`terminal_weblinks` Whether web links displayed in the Terminal tab are made clickable.

`show_rmd_render_command` Whether to print the render command use to knit R Markdown documents in the R Markdown tab.

`enable_text_drag` Whether to enable moving text on the editing surface by clicking and dragging it.

`show_hidden_files` Whether to show hidden files in the Files pane.

`always_shown_files` List of file names (case sensitive) that are always shown in the Files Pane, regardless of whether hidden files are shown

`always_shown_extensions` List of file extensions (beginning with ., not case sensitive) that are always shown in the Files Pane, regardless of whether hidden files are shown

`sort_file_names_naturally` Whether to sort file names naturally, so that e.g., file10.R comes after file9.R

`sync_files_pane_working_dir` Whether to change the directory in the Files pane automatically when the working directory in R changes.

`jobs_tab_visibility` The visibility of the Jobs tab.

`show_launcher_jobs_tab` Whether to show the Workbench Jobs tab in RStudio Pro and RStudio Workbench.

`launcher_jobs_sort` How to sort jobs in the Workbench Jobs tab in RStudio Pro and RStudio Workbench.

`busy_detection` How to detect busy status in the Terminal.

`busy_exclusion_list` A list of apps that should not be considered busy in the Terminal.

`knit_working_dir` The working directory to use when knitting R Markdown documents.

`doc_outline_show` Which objects to show in the document outline pane.

`latex_preview_on_cursor_idle` When to preview LaTeX mathematical equations when cursor has not moved recently.

`wrap_tab_navigation` Whether to wrap around when going to the previous or next editor tab.

`global_theme` The theme to use for the main RStudio user interface.

`git_diff_ignore_whitespace` Whether to ignore whitespace when generating diffs of version controlled files.

`console_double_click_select` Whether double-clicking should select a word in the Console pane.

`console_suspend_blocked_notice` Whether the 'Auto Suspension Blocked' icon should appear in the R Console toolbar.

`console_suspend_blocked_notice_delay` How long to wait before warning that automatic session suspension has been paused. Higher values for less frequent notices.

`new_proj_git_init` Whether a git repo should be initialized inside new projects by default.

`new_proj_use_renv` Whether an renv environment should be created inside new projects by default.

`root_document` The root document to use when compiling PDF documents.

`show_user_home_page` When to show the server home page in RStudio Workbench.

`reuse_sessions_for_project_links` Whether to reuse sessions when opening projects in RStudio Workbench.

`vcs_enabled` Whether to enable RStudio's version control system interface.

`vcs_autorefresh` Automatically refresh VCS status?

`git_exe_path` The path to the Git executable to use.

`svn_exe_path` The path to the Subversion executable to use.

`terminal_path` The path to the terminal executable to use.

`rsa_key_path` The path to the SSH key file to use.

`ssh_key_type` The encryption type to use for the SSH key file.

`use_devtools` Whether to use the devtools R package.

`clean_before_install` Always use `-preclean` when installing package.

`use_internet2` Whether to use Internet2 for networking on R for Windows.

`use_secure_download` Whether to use secure downloads when fetching R packages.

`cleanup_after_r_cmd_check` Whether to clean up temporary files after running R CMD CHECK.

`view_dir_after_r_cmd_check` Whether to view the directory after running R CMD CHECK.

`hide_object_files` Whether to hide object files in the Files pane.

`restore_last_project` Whether to restore the last project when starting RStudio.

`project_safe_startup_seconds` The number of seconds after which a project is deemed to have successfully started.

`use_tinytex` Use tinytex to compile .tex files.

`clean_texi2dvi_output` Whether to clean output after running Texi2Dvi.

`latex_shell_escape` Whether to enable shell escaping with LaTeX documents.

`restore_project_r_version` Whether to restore the last version of R used by the project in RStudio Pro and RStudio Workbench.

`clang_verbose` The verbosity level to use with Clang (0 - 2)

`submit_crash_reports` Whether to automatically submit crash reports to RStudio.

`default_r_version` The R version to use by default.

`data_viewer_max_columns` The maximum number of columns to show at once in the data viewer.

`data_viewer_max_cell_size` The maximum number of characters to show in a data viewer cell.

`enable_screen_reader` Support accessibility aids such as screen readers.

`typing_status_delay_ms` Number of milliseconds to wait after last keystroke before updating live region.

`reduced_motion` Reduce use of animations in the user interface.

`tab_key_move_focus` Tab key moves focus out of text editing controls instead of inserting tabs.

`find_panel_legacy_tab_sequence` In source editor find panel, tab key moves focus directly from find text to replace text.

`show_focus_rectangles` Control with keyboard focus displays a visual focus indicator.

`show_panel_focus_rectangle` Show which panel contains keyboard focus.

`auto_save_on_idle` How to deal with changes to documents on idle.

`auto_save_idle_ms` The idle period, in milliseconds, after which documents should be auto-saved.

`auto_save_on_blur` Whether to automatically save when the editor loses focus.

`terminal_initial_directory` Initial directory for new terminals.

`full_project_path_in_window_title` Whether to show the full path to project in desktop window title.

`visual_markdown_editing_is_default` Whether to enable visual editing by default for new markdown documents

`visual_markdown_editing_list_spacing` Default spacing for lists created in the visual editor

`visual_markdown_editing_wrap` Whether to automatically wrap text when writing markdown

`visual_markdown_editing_wrap_at_column` The column to wrap text at when writing markdown

`visual_markdown_editing_references_location` Placement of footnotes within markdown output.

`visual_markdown_editing_canonical` Whether to write canonical visual mode markdown when saving from source mode.

`visual_markdown_editing_max_content_width` Maximum content width for visual editing mode, in pixels

`visual_markdown_editing_show_doc_outline` Whether to show the document outline by default when opening R Markdown documents in visual mode.

`visual_markdown_editing_show_margin` Whether to show the margin guide in the visual mode code blocks.

`visual_markdown_code_editor_line_numbers` Whether to show line numbers in the code editors used in visual mode

`visual_markdown_editing_font_size_points` The default visual editing mode font size, in points

`visual_markdown_code_editor` The name of the editor to use to provide code editing in visual mode

`zotero_libraries` Zotero libraries to insert citations from.

`emoji_skintone` Preferred emoji skintone

`disabled_aria_live_announcements` List of aria-live announcements to disable.

`screenreader_console_announce_limit` Maximum number of lines of console output announced after a command.

`file_monitor_ignored_components` List of path components; file monitor will ignore paths containing one or more of these components.

`install_pkg_deps_individually` Whether to install R package dependencies one at a time.

`graphics_backend` R graphics backend.

graphics_antialiasing Type of anti-aliasing to be used for generated R plots.

browser_fixed_width_fonts List of fixed-width fonts to check for browser support.

python_type The Python type.

python_version The Python version.

python_path The path to the default Python interpreter.

save_retry_timeout The maximum amount of seconds of retry for save operations.

insert_native_pipe_operator Whether the Insert Pipe Operator command should use the native R pipe operator, |>

command_palette_mru Whether to keep track of recently used commands in the Command Palette

show_memory_usage Whether to compute and show memory usage in the Environment Pane

memory_query_interval_seconds How many seconds to wait between automatic queries of memory statistics (0 to disable)

terminal_python_integration Enable Python terminal hooks. When enabled, the RStudio-configured version of Python will be placed on the PATH.

session_protocol_debug Enable session protocol debug logging showing all session requests and events

python_project_environment_automatic_activate When enabled, if the active project contains a Python virtual environment, then RStudio will automatically activate this environment on startup.

check_null_external_pointers When enabled, RStudio will detect R objects containing null external pointers when building the Environment pane, and avoid introspecting their contents further.

ui_language The IDE's user-interface language.

native_file_dialogs Whether RStudio Desktop will use the operating system's native File and Message dialog boxes.

discard_pending_console_input_on_error When enabled, any pending console input will be discarded when an (uncaught) R error occurs.

editor_scroll_multiplier An integer value, 1-200, to set the editor scroll multiplier. The higher the value, the faster the scrolling.

text_rendering Control how text is rendered within the IDE surface.

disable_renderer_accessibility Disable Electron accessibility support.

copilot_enabled When enabled, RStudio will use GitHub Copilot to provide code suggestions.

copilot_completions_delay The delay (in milliseconds) before GitHub Copilot completions are requested after the cursor position has changed.

Source

<https://github.com/rstudio/rstudio/raw/main/src/cpp/session/resources/schema/user-prefs-schema.json>

```
prefs_rstudio_v      prefs_rstudio_v
```

Description

The RStudio User Preferences, their description and default value for released versions of RStudio.

Usage

```
prefs_rstudio_v
```

Format

A list with preferences from 10 released versions of RStudio.

Source

<https://github.com/rstudio/rstudio/raw/main/src/cpp/session/resources/schema/user-prefs-schema.json>

See Also

[prefs_rstudio](#)

```
prefs_schema      Look up RStudio Preference Schema
```

Description

Finds and downloads the RStudio user preference schema for your current version of the IDE (or the most recent released version of the IDE). The preference details for all released versions of the IDE are included in the **rsprefs** package. The preferences from the latest version are available in [prefs_rstudio](#), but note that those preferences may not match your version of RStudio unless you're using the latest preview version available at <https://dailies.rstudio.com>.

Usage

```
prefs_schema(version = NULL, quiet = FALSE)
```

Arguments

version	The version of the RStudio IDE, e.g. "1.3.959", "1.4.1717", or "2021.09.0+351".
quiet	Suppress console messages and output

Examples

```
prefs_schema("2021.09.0+351")
```

Description

Save named snapshots of your RStudio preferences that you can apply later.

- `snapshot_prefs_save()`: Save your RStudio user preferences with a name that you can later use to identify these particular settings.
- `snapshot_prefs_use()`: Apply a saved snapshot.
- `snapshot_prefs_list()`: List available snapshots.
- `snapshot_prefs_undo()`: Undo the last applied snapshot.

Usage

```
snapshot_prefs_save(  
  name,  
  path = NULL,  
  include = NULL,  
  exclude = NULL,  
  source = "user",  
  exclude_os_prefs = TRUE,  
  overwrite = FALSE,  
  preview = FALSE  
)
```

```
snapshot_prefs_use(  
  name = NULL,  
  path = NULL,  
  exclude_os_prefs = TRUE,  
  verbose = FALSE,  
  preview = FALSE  
)
```

```
snapshot_prefs_list(path = NULL, verbose = TRUE)
```

```
snapshot_prefs_undo(verbose = TRUE)
```

Arguments

- | | |
|------|--|
| name | The name of the snapshot to save or apply. <ul style="list-style-type: none">• In <code>snapshot_prefs_save()</code>, set to <code>NULL</code> to show the preferences that would be included in the snapshot.• In <code>snapshot_prefs_use()</code>, set to <code>NULL</code> to list available snapshots. |
|------|--|

path	A GitHub gist ID or local path where the snapshot should be saved. To create a new public gist, set path = "new gist". For a new private gist, use path = "new private gist".
include	Names of RStudio preferences to include. If provided, only these preferences are included. See prefs_rstudio for preference names.
exclude	Names of RStudio preferences to exclude from the snapshot. See prefs_rstudio for all of the preference names.
source	The source of the current preference value. Preferences are set at different levels, from lowest to highest precedence: <ul style="list-style-type: none"> • "default" are RStudio's built-in defaults • "computed" are detected or supplied from external sources, e.g. the path to git on your system • "system" are derived from system-wide rstudio-prefs.json • "user" are set by the user for all sessions (global options) • "project" are set by the current project session <p>The default is "user", since these are the settings you set yourself for all projects. You can include any number of sources, or "all" to include all preferences regardless of source.</p>
exclude_os_prefs	Excludes operating-system or machine-dependent system preferences from the snapshot or the snapshot restore.
overwrite	If the snapshot exists, should it be overwritten?
preview	When TRUE, previews the preferences that will be or are included in the snapshot, but does not save or apply them.
verbose	Prints or suppress informative output

Examples

```
if (interactive()) {
  tmpfile <- tempfile(fileext = ".json")
  snapshot_prefs_save("example", path = tmpfile)
  snapshot_prefs_list(tmpfile)
}
```

Index

* datasets

prefs_rstudio, [3](#)

prefs_rstudio_v, [12](#)

prefs_reset_defaults, [2](#)

prefs_rstudio, [2](#), [3](#), [12](#), [14](#)

prefs_rstudio_v, [12](#)

prefs_schema, [12](#)

snapshot_prefs, [13](#)

snapshot_prefs_list (snapshot_prefs), [13](#)

snapshot_prefs_save, [2](#)

snapshot_prefs_save (snapshot_prefs), [13](#)

snapshot_prefs_undo (snapshot_prefs), [13](#)

snapshot_prefs_undo(), [2](#)

snapshot_prefs_use (snapshot_prefs), [13](#)