Package: syncr (via r-universe)

September 19, 2024

Title Synchronise Directories

Version 0.0.3

Author Rich FitzJohn

Maintainer Rich FitzJohn <rich.fitzjohn@gmail.com>

Description Syncronise directories using rsync.

License MIT + file LICENSE

LazyData true

URL https://github.com/mrc-ide/syncr

BugReports https://github.com/mrc-ide/syncr/issues

RoxygenNote 5.0.1

Suggests testthat

Repository https://mrc-ide.r-universe.dev

RemoteUrl https://github.com/mrc-ide/syncr

RemoteRef master

RemoteSha 09d2cdae3c97538cc37c2605019c65b15b7316c1

Contents

has_rsync *Check if there is an rsync we can use*

Description

Check if there is an rsync we can use.

Usage

```
has_rsync()
```

rsync *Wrapper around rsync*

Description

Wrapper around all of *rsync*'s options. I'm not sure how useful this will be (because you'll really need to read rsync's extensive manual to use this, but perhaps it will be useful.

Usage

```
rsync(src, dest, verbose = FALSE, quiet = FALSE, no_motd = FALSE,
  checksum = FALSE, archive = FALSE, recursive = FALSE,
  relative = FALSE, no_implied_dirs = FALSE, backup = FALSE,
  backup_dir = NULL, suffix = NULL, update = FALSE, inplace = FALSE,
  append = FALSE, dirs = FALSE, links = FALSE, copy_links = FALSE,
  copy_unsafe_links = FALSE, safe_links = FALSE, copy_dirlinks = FALSE,
  keep_dirlinks = FALSE, hard_links = FALSE, perms = FALSE,
  executability = FALSE, chmod = NULL, owner = FALSE, group = FALSE,
  devices = FALSE, specials = FALSE, times = FALSE,
  omit_dir_times = FALSE, super = FALSE, sparse = FALSE,
  dry_run = FALSE, whole_file = FALSE, one_file_system = FALSE,
  block_size = NULL, rsh = NULL, rsync_path = NULL, existing = FALSE,
  ignore_existing = FALSE, remove_source_files = FALSE, delete = FALSE,
  delete_before = FALSE, delete_during = FALSE, delete_after = FALSE,
  delete_excluded = FALSE, ignore_errors = FALSE, force = FALSE,
  max_delete = NULL, max_size = NULL, min_size = NULL, partial = FALSE,
  partial_dir = NULL, delay_updates = FALSE, prune_empty_dirs = FALSE,
  numeric_ids = FALSE, timeout = NULL, ignore_times = FALSE,
  size_only = FALSE, modify_window = NULL, temp_dir = NULL,
  fuzzy = FALSE, compare_dest = NULL, copy_dest = NULL,
  link_dest = NULL, compress = FALSE, compress_level = NULL,
  cvs_exclude = FALSE, filter = NULL, exclude = NULL,
  exclude_from = NULL, include = NULL, include_from = NULL,
  files_from = NULL, from0 = FALSE, address = NULL, port = NULL,
```

```
  sockopts = NULL, blocking_io = FALSE, stats = FALSE,
  eight_bit_output = FALSE, human_readable = FALSE, progress = FALSE,
  itemize_changes = FALSE, out_format = NULL, log_file = NULL,
  log_file_format = NULL, password_file = NULL, list_only = FALSE,
  bwlimit = NULL, write_batch = NULL, only_write_batch = NULL,
  read_batch = NULL, protocol = NULL, checksum_seed = NULL,
  ipv4 = FALSE, ipv6 = FALSE, extended_attributes = FALSE,
  cache = FALSE, drop_src_directory = FALSE, args_only = FALSE)
```

Arguments

src	Source files to copy. See details
dest	A single destination path, possibly remote.
verbose	increase verbosity
quiet	suppress non-error messages
no_motd	suppress daemon-mode MOTD (see caveat)
checksum	skip based on checksum, not mod-time & size
archive	archive mode; same as recursive, links, perms, times, group, owner, devices specials but no hard_links
recursive	recurse into directories
relative	use relative path names
no_implied_dirs	don't send implied dirs with relative
backup	make backups (see suffix & backup_dir)
backup_dir	make backups into hierarchy based in DIR
suffix	backup suffix (default ~ w/o backup_dir)
update	skip files that are newer on the receiver
inplace	update destination files in-place
append	append data onto shorter files
dirs	transfer directories without recursing
links	copy symlinks as symlinks
copy_links	transform symlink into referent file/dir
copy_unsafe_links	only "unsafe" symlinks are transformed
safe_links	ignore symlinks that point outside the tree
copy_dirlinks	transform symlink to dir into referent dir
keep_dirlinks	treat symlinked dir on receiver as dir
hard_links	preserve hard links
perms	preserve permissions
executability	preserve executability
chmod	affect file and/or directory permissions

`owner`	preserve owner (super-user only)
`group`	preserve group
`devices`	preserve device files (super-user only)
`specials`	preserve special files
`times`	preserve times
`omit_dir_times`	omit directories when preserving times
`super`	receiver attempts super-user activities
`sparse`	handle sparse files efficiently
`dry_run`	show what would have been transferred
`whole_file`	copy files whole (without rsync algorithm)
`one_file_system`	don't cross filesystem boundaries
`block_size`	force a fixed checksum block-size
`rsh`	specify the remote shell to use
`rsync_path`	specify the rsync to run on remote machine
`existing`	skip creating new files on receiver
`ignore_existing`	skip updating files that exist on receiver
`remove_source_files`	sender removes synchronized files (non-dir)
`delete`	delete extraneous files from dest dirs
`delete_before`	receiver deletes before transfer (default)
`delete_during`	receiver deletes during xfer, not before
`delete_after`	receiver deletes after transfer, not before
`delete_excluded`	also delete excluded files from dest dirs
`ignore_errors`	delete even if there are I/O errors
`force`	force deletion of dirs even if not empty
`max_delete`	don't delete more than NUM files
`max_size`	don't transfer any file larger than SIZE
`min_size`	don't transfer any file smaller than SIZE
`partial`	keep partially transferred files
`partial_dir`	put a partially transferred file into DIR
`delay_updates`	put all updated files into place at end
`prune_empty_dirs`	prune empty directory chains from file_list
`numeric_ids`	don't map uid/gid values by user/group name
`timeout`	set I/O timeout in seconds
`ignore_times`	don't skip files that match size and time

`size_only`	skip files that match in size
`modify_window`	compare mod_times with reduced accuracy
`temp_dir`	create temporary files in directory DIR
`fuzzy`	find similar file for basis if no dest file
`compare_dest`	also compare received files relative to DIR
`copy_dest`	... and include copies of unchanged files
`link_dest`	hardlink to files in DIR when unchanged
`compress`	compress file data during the transfer
`compress_level`	explicitly set compression level
`cvs_exclude`	auto_ignore files in the same way CVS does
`filter`	add a file_filtering RULE
`exclude`	exclude files matching PATTERN
`exclude_from`	read exclude patterns from FILE
`include`	don't exclude files matching PATTERN
`include_from`	read include patterns from FILE
`files_from`	read list of source-file names from FILE
`from0`	all *from/filter files are delimited by 0s
`address`	bind address for outgoing socket to daemon
`port`	specify double_colon alternate port number
`sockopts`	specify custom TCP options
`blocking_io`	use blocking I/O for the remote shell
`stats`	give some file-transfer stats
`eight_bit_output`	leave high-bit chars unescaped in output
`human_readable`	output numbers in a human-readable format
`progress`	show progress during transfer
`itemize_changes`	output a change-summary for all updates
`out_format`	output updates using the specified FORMAT
`log_file`	log what we're doing to the specified FILE
`log_file_format`	log updates using the specified FMT
`password_file`	read password from FILE
`list_only`	list the files instead of copying them
`bwlimit`	limit I/O bandwidth; KBytes per second
`write_batch`	write a batched update to FILE
`only_write_batch`	like write_batch but w/o updating dest

read_batch	read a batched update from FILE
protocol	force an older protocol version to be used
checksum_seed	set block/file checksum seed (advanced)
ipv4	prefer IPv4
ipv6	prefer IPv6
extended_attributes	copy extended attributes, resource forks
cache	disable fcntl(F_NOCACHE)
drop_src_directory	In the case where src is a single directory, don't copy the directory, but copy the contents.
args_only	Don't run anything and instead return the arguments that would have been passed to rsync.

---

syncr *syncr*

Description

Wrapper around rsync, with only a few options exposed. *WARNING*: This command modifies files on your computer and therefore is potentially dangerous. Files in dest may be overwritten or deleted!

Usage

```
syncr(src, dest, archive = TRUE, compress = TRUE, verbose = FALSE,
  relative = FALSE, delete = FALSE, dry_run = FALSE,
  drop_src_directory = FALSE, inplace = NULL, args_only = FALSE)
```

Arguments

src	Source files to copy. See details
dest	A single destination path, possibly remote.
archive	Copy in archive mode, being recursive, preserving attributes etc? Generally this is wanted.
compress	Compress files on transfer?
verbose	Be verbose? (Always TRUE when using dry_run).
relative	Copy relative paths only, rather than the last part of the file name. So if you have a directory src and you want to syncronise all the .c files in it, you could use src="src/*.c", relative=TRUE. You can include a dot at any point in a path name to indicate where the relative directory should start from; e.g. /absolute/path/to/./src/*.c which will create a directory src with .c files in it.

`delete` Delete files on the remote machine. When mirroring this is generally what you want to do but it is not enabled by default because it's potentially dangerous.

`dry_run` Don't actually do anything, but print what would be done instead.

`drop_src_directory`
In the case where `src` is a single directory, don't copy the directory, but copy the contents.

`inplace` Copy files in place. This helps when copying to windows network shares where the unix emulation layer and windows don't get on very well. The downside is that if the copy fails it leaves files in an inconsistent state. By default, it will be used on windows and when the destination is an absolute path.

`args_only` Don't run anything and instead return the arguments that would have been passed to rsync.

Details

This is a simple wrapper around a few of the most common arguments to `rsync`. For a more complete interface, see `rsync`.

Index