

ТРАНСФЕР ДАННЫХ

Один раздел на всё: папка на этой машине, сетевая шара, сервер по SFTP, облако. Раньше их было два — Robocopy отдельно, RClone отдельно, — и чтобы выбрать кнопку, надо было сперва знать, какой инструмент куда дотягивается. Теперь выбирают не инструмент, а действие: что сделать и куда отправить. Движок выбирается сам и назван в окне.

Пять операций

Операция	Что происходит с назначением	rclone	CLI аудитора
Односторонняя	новое приезжает, лишнее остаётся жить	copy	sync
Зеркало	назначение становится копией источника, лишнее удаляется	sync	backup
Перенести	приезжает и пропадает с источника	move	—
Двусторонняя	обе стороны догоняют друг друга	bisync	sync2
Сверить	ничего не пишется, только сравнение	check	compare

Зеркало и Односторонняя — не одно и то же с разной галочкой удаления.

У зеркала назначение — копия и ничего своего не имеет. Так работает эталонная машина и её реплики: изменилось на эталоне — разъехалось по всем, включая удаления. Это самый частый режим для файлов.

У односторонней у назначения своя жизнь. Что-то легло на одно облако и должно доехать до другого, где уже лежит своё. Сделать второе облако точной копией первого — значит стереть то, что там было.

У операции «Перенести» нет команды в CLI аудитора. Кнопка там не предлагается, а если позвать напрямую — будет ошибка. Молчаливая подмена переноса на зеркало обошлась бы дороже отказа.

Кто везёт

Выбор не спрашивают, его показывают в окне строкой «движок и почему».

RClone берётся, когда в паре есть облако или сохранённое подключение, когда адрес — ссылка `http(s)://`, когда операция двусторонняя, и когда сверяют по хешу.

Robocopy берётся на всё остальное — папка в папку, папка в шару. Он не убран за ненадобностью, он понижен из раздела в движок: на дереве папок он быстрее, и он один из двух переносит NTFS-атрибуты, которые rclone на Windows теряет.

Robocopy отвечает битовой маской, а не привычным «ноль или ошибка»: `0..7` — это успех, `8` и выше — сбой. Код `1` означает «файлы скопированы» и встречается чаще всех. Обёртки, проверяющие `!= 0`, объявляют ошибкой каждое удачное копирование.

Один эталон, список машин

Назначение — не одно поле, а список. Разложить с эталонной машины на четыре реплики — это один запуск, а не четыре.

Вся раскладка собирается до того, как тронется первый байт: для каждой машины уже известно, каким движком поедет, и отказ виден сразу, а не на середине. Если пара невозможна — например, копирование папки в папку просят сделать через облако, — это будет сказано до старта.

Маски

Маски берутся из тех же групп расширений, что и везде в программе. `docx` и `*.docx` — одно и то же, точка внутри означает имя файла целиком: `Thumbs.db` останется `Thumbs.db`, а не превратится в `*.thumbs.db`. Регистр не различается, `*.DOCX` рядом с `*.docx` не удвоится.

Отрицательные маски — то, что не должно уехать. Robocopy получает их через `/XF`. У rclone включения и исключения не складываются: если задан хоть один `--include`, всё остальное и так отсечено, и исключения не передаются.

Начиная с 12 масок список уходит в файл через `--filter-from`, а не выкладывается в командную строку. В файле сперва исключения, потом включения, в конце `- *`.

Проверено: 301 маска в Robocopy работает. Ограничение в 128 относится к `/MT` — это число потоков, а не масок.

У `rclone` `*.docx` ловит файл на любой глубине. `**/*.docx` — не то же самое: он пропускает файлы верхнего уровня.

Пробный прогон

Галочка «Пробный прогон» есть у всех операций и ничего не пишет. Robocopy получает `/L` `/BYTES /FP`, `rclone` — `--dry-run`.

Пробные прогоны намеренно устроены дешевле настоящих: их бывает много, и они не должны стоить столько же, сколько работа.

Сначала упаковать

Мелкие файлы по сети идут медленно, и виновата не скорость канала. Замерено на живом сервере через SFTP:

- один файл — 24 МБ/с;
- триста мелких — 0.46 МБ/с.

Разница в пятьдесят раз, при загрузке процессора 19.7%. В процессор не упирается, значит больше ядер не помогут; время уходит на то, что сеть здороваются с каждым файлом отдельно. `-sftp-concurrency` тут тоже не помогает: он про параллельность внутри одного файла.

Помогает одно — отправить один файл вместо трёхсот. Переключатель «Сначала упаковать» собирает источник в архивы, кладёт их в `workspace\transfer_pack\<имя>` и отправляет уже оттуда. Если упаковать нечего — источник не на этой машине, а на удалённой стороне — будет сказано об этом, а не тихий обычный перенос.

Потолок удалений

`--max-delete` доступен и не навязан. Значения по умолчанию нет: это была бы догадка о том, как человек работает, а релиз, в котором отваливается сотня старых файлов, — обычное дело.

Двусторонняя память

`bisync` помнит прошлый прогон списком файлов. `rclone` держал бы его в `%LOCALAPPDATA%`; здесь он лежит в `config\bisync`, вместе с проектом.

Не в `workspace`: чистка `workspace` опустошает папку, и потеря этих списков ломается тихо — следующий запуск потребует полной пересинхронизации, то есть ровно того единственного момента, когда двусторонней синхронизации надо объяснить, какая сторона главная.

Что рядом

- [RCLONE_OPERATIONS_RU.md](#) — подключения, хранилище, диагностика, шифрование конфига.
- [RCLONE_PORTABILITY_RU.md](#) — переносимость: конфиг и ключи, которые едут вместе с папкой.
- [ARCHIVE_OPERATIONS_RU.md](#) — архивация сама по себе.
- [SYNC_PROFILES_RU.md](#) — профили пар: отдельная, более старая дорога через `config\sync_pairs.json`.