

RClone: установка, конфиги и портативность

Этот документ объясняет механику RClone в Audion Disk Tools: где лежит `rclone.exe`, где лежит `rclone.conf`, что делает импорт/экспорт, и почему `Custom path` не является отдельной установкой.

Главное правило: **binary install** и **config scope** - это две разные оси.

Коротко

`Installer RClone` отвечает только за `rclone.exe`.

`Configuration` отвечает за выбор и перенос `rclone.conf`.

`Custom path` - это не `Custom install`. Это явный путь к конкретному `rclone.conf`.

Почему нет Custom install

В GUI нет третьего режима установки `Custom install`.

Если раньше в обсуждении звучало слово `Custom`, правильный смысл такой:

`Custom path` = использовать указанный пользователем файл `rclone.conf`

Это не меняет расположение `rclone.exe`, не скачивает отдельный RClone, не ставит программу в произвольную папку и не делает миграцию конфигов. Это только добавляет к командам RClone явный параметр:

```
--config "E:\path\to\rclone.conf"
```

Две независимые оси

Ось	Варианты	Что выбирает
Installer RClone	System, Portable	Куда установить или обновить <code>rclone.exe</code> .
Configuration / Scope	Portable, System, Custom path	Какой <code>rclone.conf</code> использовать при запуске команд.

Эти оси не обязаны совпадать. Например, можно иметь portable `Tools\rclone\rclone.exe`, но временно выбрать `Configuration / System`, чтобы команда использовала системный `%APPDATA%\rclone\rclone.conf`.

Installer RClone

System

Кнопка `System` запускает:

```
install\Install-System-Rclone.cmd
```

Скрипт скачивает latest stable Windows AMD64 RClone, распаковывает его и устанавливает user-scope binary в:

```
%LOCALAPPDATA%\Programs\rclone\rclone.exe
```

Затем добавляет эту папку в user `PATH` и проверяет:

```
rclone.exe version
```

Важно: `System` installer не требует глобальной установки в `Program Files`, не должен трогать UAC и не перезаписывает:

```
%APPDATA%\rclone\rclone.conf
```

Portable

Кнопка `Portable` запускает:

```
install\Install-Portable-Rclone.cmd
```

Скрипт скачивает latest stable Windows AMD64 RClone, распаковывает его и кладёт portable binary в:

```
Tools\rclone\rclone.exe
```

Он создаёт служебные папки:

```
config\rclone  
tmp\rclone-cache
```

Но не перезаписывает:

```
config\rclone\rclone.conf
```

То есть обновление portable binary не покрывает авторизации, OAuth tokens и remote-настройки.

Как выбирается rclone.exe

Audion Disk Tools ищет `rclone.exe` в фиксированном порядке:

1. `Tools\rclone\rclone.exe`
2. `tools\rclone\rclone.exe`
3. `runtime\rclone\rclone.exe`
4. `runtime\tools\rclone\rclone.exe`
5. `system_core\rclone.exe`
6. `rclone.exe` из `PATH`
7. `rclone` из `PATH`

Если portable binary существует в `Tools\rclone`, GUI обычно будет использовать его первым.

`System` install полезен как системный инструмент, как fallback через `PATH` и для обычного терминала Windows. Текущий активный executable показывается в верхней строке RClone-раздела и в command preview.

Configuration / Scope

Scope находится внутри блока **Configuration**, потому что это не установка, а выбор файла конфигурации. Он определяет, какой **rclone.conf** получит команда.

Portable

Это дефолтный режим Audion Disk Tools.

Команды получают:

```
--config "<ROOT>\config\rclone\rclone.conf"  
--cache-dir "<ROOT>\tmp\rclone-cache"
```

Плюсы:

- конфиг живёт рядом с программой;
- перенос проекта на другую машину сохраняет remotes вместе с проектом;
- OAuth tokens и SFTP remotes не попадают в JSON/YAML проекта;
- **rclone.conf** исключён из Git/release-шума.

Минусы:

- если внутри **rclone.conf** есть абсолютные пути к ключам, service account JSON, **known_hosts** или локальным папкам, после переноса их нужно проверить;
- перенос OAuth tokens между машинами зависит от backend-а и политики провайдера.

System

В этом режиме GUI не добавляет **--config**.

RClone использует стандартный системный конфиг:

```
%APPDATA%\rclone\rclone.conf
```

Это удобно, если RClone уже давно настроен в Windows, и пользователь хочет использовать существующие remotes без импорта.

Custom path

В этом режиме пользователь явно указывает файл **rclone.conf**.

Команды получают:

```
--config "<custom path>"
```

Custom path нужен для диагностики, временного подключения чужого конфига, тестовой конфигурации или отдельного защищённого расположения. Это не установка RClone.

Configuration

Линейка **Configuration** работает с файлами конфигурации, а не с binary. В ней находятся **Portable**, **System**, **Custom path**, поле для custom-файла и действия **Import**, **Export**, **Doctor**.

Import

Import копирует системный конфиг:

```
%APPDATA%\rclone\rclone.conf
```

в portable config:

```
config\rclone\rclone.conf
```

Если portable config уже есть, GUI сначала создаёт backup рядом с ним.

Используйте **Import**, если на машине уже настроен RClone, а проект нужно сделать самостоятельное.

Export

Export копирует portable config:

```
config\rclone\rclone.conf
```

в системный config:

```
%APPDATA%\rclone\rclone.conf
```

Если system config уже есть, GUI сначала создаёт backup рядом с ним.

Используйте **Export**, если remotes из Audion Disk Tools нужны в обычном системном RClone.

Doctor

Doctor ничего не меняет.

Он read-only проверяет:

- активный **Configuration / Scope**;
- активный путь к **rclone.conf**;
- portable cache path;
- список remotes через **rclone listremotes**;
- подозрительные абсолютные пути внутри **rclone.conf**.

Doctor нужен перед переносом на другую машину, после обновления RClone или если remote внезапно перестал работать.

Что происходит на чистой машине

Сценарий с чистым проектом из GitHub:

1. Пользователь запускает GUI.
2. В **ОБЛАЧНЫЕ ОПЕРАЦИИ** нажимает **Portable** или **System**.
3. Устанавливается только **rclone.exe**.
4. Конфиги не накрываются и не создаются с чужими секретами.
5. По умолчанию выбран **Configuration / Portable**.
6. При первом OAuth/SFTP setup RClone создаст или обновит:

```
config\rclone\rclone.conf
```

Пользователю не нужно отдельно нажимать кнопку, чтобы portable config начал работать.

Дефолтный **Configuration / Portable** уже добавляет **--config** к командам.

Что происходит при обновлении RClone

Обновление **Portable** заменяет **Tools\rclone**, но не трогает:

```
config\rclone\rclone.conf
```

Обновление **System** заменяет user-scope binary в:

```
%LOCALAPPDATA%\Programs\rclone
```

и не трогает:

```
%APPDATA%\rclone\rclone.conf
```

То есть обновление binary и хранение remote-настроек разделены. Это снижает риск рекурсии, случайного перезаписывания OAuth tokens и потери настроенных remotes.

Перенос проекта на другой путь или машину

Portable config лежит внутри проекта:

```
config\rclone\rclone.conf
```

Поэтому при переносе папки проекта команды продолжают указывать на новый путь через **<ROOT>**.

Но сам **rclone.conf** может содержать абсолютные пути, например:

```
key_file = C:\Users\User\.ssh\id_ed25519
service_account_file = E:\Secrets\google.json
known_hosts_file = C:\Users\User\.ssh\known_hosts
```

Такие пути RClone не может автоматически сделать переносимыми без риска сломать авторизацию. Поэтому перед переносом используйте **Doctor** и проверьте warning-и.

Release и cleanup

config\rclone\rclone.conf содержит secrets и не должен попадать в публичный репозиторий или release-архив.

Файл исключён через **.gitignore**:

```
config/rclone/*  
!config/rclone/.gitkeep
```

`cleanup_project.cmd` не удаляет portable `config\rclone`, но может очищать временный cache:

```
tmp\rclone-cache
```

Это нормально: cache пересоздается, а `rclone.conf` остаётся.

Практические сценарии

Хочу полностью portable-поведение

1. Нажмите `Portable`.
2. Оставьте `Configuration / Scope = Portable`.
3. Настройте remotes через `Провайдеры OAuth`, `Create/update SFTP remote` или `Config`.
4. Проверьте `List remotes`.
5. Перед переносом запустите `Doctor`.

Хочу использовать уже настроенный системный RClone

1. Нажмите `System`, если `rclone.exe` ещё не установлен в Windows.
2. Выберите `Configuration / Scope = System`.
3. Проверьте `List remotes`.
4. Если всё работает, можно оставить так или нажать `Import`, чтобы перенести system config в portable.

Хочу перенести remotes из системы в проект

1. Выберите `Configuration / Import`.
2. Оставьте `Configuration / Scope = Portable`.
3. Запустите `Doctor`.
4. Проверьте remotes через `List remotes`.

Хочу отдать remotes из проекта в обычный RClone

1. Выберите **Configuration / Export**.
2. Запустите обычный терминал Windows.
3. Проверьте:

```
rclone listremotes
```

Хочу временно проверить чужой rclone.conf

1. Выберите **Configuration / Scope = Custom path**.
2. Укажите путь к файлу **rclone.conf**.
3. Запустите **List remotes** или **Doctor**.

После проверки верните **Configuration / Scope = Portable**, если это основной режим проекта.